14:34:05 OCA PAD AMENDMENT - PROJECT HEADER INFORMATION 03/12/92

Active

Project #: E-16-A03 Cost share #: Rev #: 14
Center # : R6483-0A0 Center shr #: OCA file #:
Work type : RES
Contract#: NAG 3-895 Mod #: ADMIN. Document : GRANT
Prime #: Contract entity: GTRC

Subprojects ? : Y CFDA: 43.002
Main project #: PE #: N/A

Project unit: AERO ENGR Unit code: 02.010.110
Project director(s):
KAMAT M P AERO ENGR (404)894-7439

Sponsor/division names: NASA / LEWIS RESEARCH CTR, OH
Sponsor/division codes: 105 / 011

Award period: 880616 to 920617 (performance) 920617 (reports)

Sponsor amount	New this change	Total to date
Contract value	0.00	110,107.00
Funded	0.00	110,107.00
Cost sharing amount		0.00

Does subcontracting plan apply ?: N

Title: PARALLEL PROCESSING FOR ANALYSIS OF TURBO MACHINERY BLADED-DISK ASSEMBLIES

PROJECT ADMINISTRATION DATA

OCA contact: Ina R. Lashley 894-4820

Sponsor technical contact	Sponsor issuing office
DAVID C JANETZKE, TECHNICAL OFFICER (216)433-6041	LORENE ALBERGOTTIE, GRANTS OFFICER (216)433-2781
MAIL STOP 23-3 NASA, LEWIS RESEARCH CENTER STRUCTURAL ANALYSIS BRANCH 21000 BROOKPARK ROAD CLEVELAND, OH 44135	MAIL STOP 500 NASA,LEWIS RESEARCH CENTER 21000 BROOKPARK ROAD CLEVELAND, OH 44135

Security class (U,C,S,TS) : U ONR resident rep. is ACO (Y/N): N
Defense priority rating : NA NA supplemental sheet
Equipment title vests with: Sponsor GIT
NONE PROPOSED.
Administrative comments -
PER NASA PROVISIONS FOR GRANTS, AT PI'S REQUEST OCA NOTIFIED NASA OF A 3-MOS. UNILATERAL TIME EXTENSION. GRANT NOW EXPIRES 6/17/92.

GEORGIA INSTITUTE OF TECHNOLOGY
OFFICE OF CONTRACT ADMINISTRATION

NOTICE OF PROJECT CLOSEOUT

Closeout Notice Date 09/08/92
Original Closeout Started 02/07/91

Project No. E-16-A03 Center No. R6483-0A0

Project Director KAMAT M P School/Lab AERO ENGR

Sponsor NASA/LEWIS RESEARCH CTR, OH

Contract/Grant No. NAG 3-895 Contract Entity GTRC

Prime Contract No.

Title PARALLEL PROCESSING FOR ANALYSIS OF TURBO MACHINERY BLADED-DISK ASSEMBLIE

Effective Completion Date 920617 (Performance) 920617 (Reports)

Closeout Actions Required:	Y/N	Date Submitted
Final Invoice or Copy of Final Invoice	Y	
Final Report of Inventions and/or Subcontracts	Y	
Government Property Inventory & Related Certificate	N	
Classified Material Certificate	N	
Release and Assignment	N	
Other	N	

Comments

Subproject Under Main Project No.

Continues Project No.

Distribution Required:

Project Director	Y
Administrative Network Representative	Y
GTRI Accounting/Grants and Contracts	Y
Procurement/Supply Services	Y
Research Property Managment	Y
Research Security Services	N
Reports Coordinator (OCA)	Y
GTRC	Y
Project File	Y
Other	N
	N

GTA

E 16-A03

INTERIM PROGRESS REPORT

NASA Grant #NAG-3-985

Period: June16, 1988 - December 31, 1989

FINITE ELEMENT CODE FOR THIN SHELL STRUCTURES ON THE ALLIANT FX/80

Manohar Kamat

Sang-youp Synn

Brian Watson

School of Aerospace Engineering

Georgia Institute of Technology

Atlanta, Georgia 30332

October 1989

Contents

1. Introduction

The objective of the research activity described in this report is to eventually provide a finite element capability for analyzing Turbo Machinery Bladed Disk Assemblies in a vector/parallel processing environment.

Analysis of aircraft turbo fan engines is computationally intensive. Problems involving aeroelastic stability and response of bladed-disk assemblies in aircraft turbo fan engines are among the most difficult problems encountered. Complications in these studies arise from the small differences between individual blades known as mistuning [1]. Previous researchers have come to believe that the static, flutter, and forced response of mistuned turbo machinery blades can be studied by analyzing each blade separately in either a pure bending or a pure torsional motion. Concurrent (parallel) processing seems to offer the greatest promise for such an analysis.

The performance limit of modern day computers with a single processing unit has been estimated at 3 billions of floating point operations per second (3 gigaflops). In view of this limit of a sequential unit, performance rates higher than 3 gigaflops can be achieved only through vectorization and/or parallelization as on Alliant FX/8. Accordingly, the efforts of this critically needed research have been geared towards developing and evaluating parallel finite element methods for static and vibration analysis of multi-degree-of-freedom models using flat thin shell elements.

Concurrent processing machines such as the FLEX-32 and Alliant FX/8 are multiple instruction, multiple data (MIMD) computers and have the potential for increasing effective calculation speeds by several orders of magnitude. But this potential increase in speed cannot be effectively utilized without the development and implementation of appropriate numerical algorithms which take advantage of the parallel computation features of this new generation of computers. Use of existing algorithms on sequential computers will not realize the full potential of these new MIMD computers and research is needed in the development of parallel structural analysis algorithms for these computers.

Issues involved in implementation of parallel structural algorithms on MIMD supercomputers are much more complex than on current sequential computers and the total hardware/software system must be taken into consideration. The implementation criteria that influence the efficiency of an algorithm include the amount of computation versus the amount of communication in a given problem, the balance of the workload among the processors, the communication paths and synchronization delays, and the size of a problem in relation to the number of processors used.

Research activity under the grant for the first year was devoted to familiarizing the two graduate students (Chris Ayers and Chung-Yul Song) with the FLEX-32 multicomputer at the CAE/CAD laboratory of the Georgia Institute of Technology.

The following tasks were completed:

1. Development of a package of linear algebra subroutines similar to LINPACK (but perhaps not as comprehensive) for operation on the FLEX-32;

2. Development and evaluation of a robust QR algorithm suitable for eigenvalue analysis and equilibrium of structural systems with real symmetric matrices. A copy of Master's Thesis Report by Chris Ayers [2] was delivered to the NASA Lewis Research Center;

(Note: During the first year of the grant, the authors had access only to the FLEX-32 facility at Georgia Tech. Versions of the software developed in items 1. and 2. on FLEX-32 cannot be easily converted for operation on Alliant FX/8 because of radical differences in the architecture of the two machines. FLEX-32 permits direct access to individual processors and synchronization, load balancing, etc. is left to the discretion of the developer. Vectorization and/or parallelization is completely automated on Alliant FX/8. So the effectiveness of the software developed in items 1. and 2. on Alliant FX/8 remains to be assessed.)

3. Development and evaluation of an elementary version of the conjugate gradient algorithm for the solution of a system of linear equations of the type commonly encountered in a finite element structural analysis;

4. For evaluation of the pay-off from the conjugate gradient algorithm of task 3 above, relative to the best known sequential algorithm, a parallel version of the $[L][D][L]^T$ algorithm was initiated. Such an algorithm is also useful for the development of a preconditioned conjugate gradient algorithm (using the element-by-element preconditioned algorithm proposed by Hughes et al.[3]) appropriate for the solution of large scale, ill-conditioned systems of linear and/or nonlinear equations often encountered in large finite element structural systems;

5. In the past six months work was initiated on parallelizing an in-house finite element code by the acronym SAPNEW [4]. The

element library of the code, which originally had an eight-to-twenty one noded isoparametric element suitable for static and transient linear analysis of solids or thick shells and plates, was replaced by a fully conforming thin flat shell element with static and vibration analysis capability. The development of this code using a $[L][D][L]^T$ factorization scheme for the solution of the linear equilibrium equations in a compacted skyline storage scheme is completed and fully validated and is being delivered to the NASA Lewis Research Center as part of this report.

In addition to the above solution scheme, preconditioned versions (using a simple diagonal preconditioner) of the global and element-by-element conjugate gradient method were completed. Even though the solution schemes need improvement by way of their performance on the Alliant, a version of SAPNEW using these solution capabilities is also being delivered.

The element-by-element preconditioned conjugate gradient algorithm for the solution of both equilibrium and eigenvalue problems should be ideal for the parallel processing environment. It is hoped that with some additional efforts, it can surpass the performance of the [L][D][L]T factorization scheme. The effectiveness of the algorithm for handling multiple load cases for static analysis as well as for extracting multiple eigenvalues and eigenvectors has also been investigated, but results are very preliminary at this time. The extension and implementation (on the Alliant) of the QR algorithm for complex matrices of the type encountered in aeroelastic analysis of the turbine blades must be undertaken next. This capability is vital to the study of flutter of mistuned turbo machinery blades.

2. Description of the SAPNEW program

To facilitate future developments of the program in the form of additional or improved capabilities, the "SAPNEW" program has been designed with the preservation of the modularity of each subroutine.

The main features of the SAPNEW program are as follows:

a) Its element library has a thin shell element[1]

. a CST (Constant Strain Triangular) element for modeling plate stretching behavior in conjunction with a fully conforming element consisting of the Q-19 element derived from four LCCT-11 elements for modeling plate bending behavior.

. The Q-19 element is reduced to the Q-12 element with 12 degrees of freedom by the static condensation procedure.

b) It permits static analysis (up to 4 load cases per execution) of shell type structures with loads which may be concentrated, distributed, or arising from thermal effects or self-weight.

c) It permits an eigen value/vector analysis with a cut-off frequency control option and a restart option.

d) It uses the sky-line storage scheme for the assembled stiffness and mass matrices, and has two options for solution of the equilibrium equations.

. The triple factorization (LDL^T) scheme with forward and

---

[1] For the details of the formulation of this element, consult reference [17].

backward substitutions;

. The global and element by element preconditioned conjugate gradient solution scheme.

3. Description of the thin,flat shell element formulation

In the local element coordinate system, the shell element behavior is synthesized from uncoupled plate-stretching and plate-bending behaviors. Coupling between these two behaviors results as a consequence of the transformation from the local to the global coordinate system.

In the "SAPNEW" program, the CST (Constant Strain Triangular) element formulation was adopted for modeling the plane-stretching behavior. The Q-12 element formulation which has 12 d.o.f. and which stems from the assemblage of four partially constrained LCCT (Linear Curvature Compatible Triangular) elements, was adopted for modeling the plate-bending behavior.

3.1 Local coordinate system, area coordinates and the transformation matrix

Let the four nodes of the quadrilateral shell element be labelled as 1,2,3,4 and those of the triangular shell element be labelled as 1,2,3. The local coordinate system is shown in Figure 1.

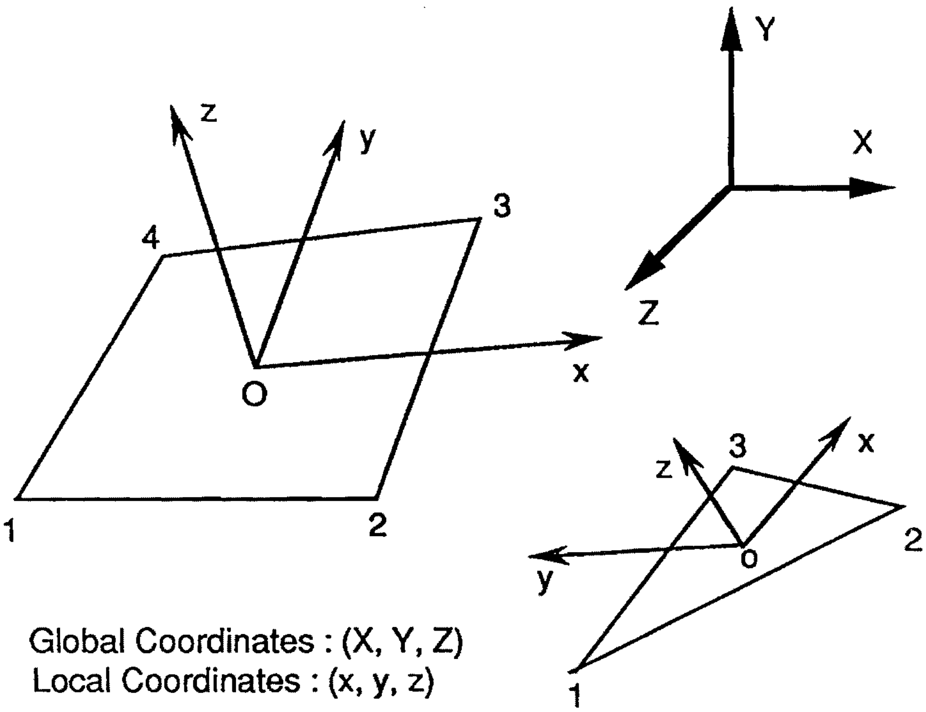


Figure 1. Local coordinate system of the shell element.

First, let the x-direction of the local coordinate system be defined as the line joining the geometric center point "O" of the element and the mid-point of the line 2-3. A unit vector x can be determined as

$$\mathbf{x} = (X_1'\,\mathbf{i} + Y_1'\,\mathbf{j} + Z_1'\,\mathbf{k}) / L_1 \tag{2.1}$$

where

$$X_1' = X_2 + X_3 - X_1 - X_4$$

$$Y_1' = Y_2 + Y_3 - Y_1 - Y_4$$

$$Z_1' = Z_2 + Z_3 - Z_1 - Z_4$$

$$L_1^2 = X_1'^2 + Y_1'^2 + Z_1'^2$$

i,j,k being unit vectors of the global coordinate system.

Likewise, let a vector **A** be defined by the line joining the point "O" and the mid-point of the line 3-4 and the vector B be defined as B = x x A. Then y-direction of the local coordinate system can be determined from B x x, and its equivalent unit vector **y** can be expressed as

$$\mathbf{y} = \{(X_3' - X_1'C)i + (Y_3' - Y_1'C)j + (Z_3' - Z_1'C)k\}/L_2 \tag{2.2}$$

where $X_3' = X_3 + X_4 - X_1 - X_2$

$Y_1' = Y_3 + Y_4 - Y_1 - Y_2$

$Z_1' = Z_3 + Z_4 - Z_1 - Z_2$

$C = (X_1'X_3' + Y_1'Y_3' + Z_1'Z_3') / L_1^2$

$X_2' = X_3' - X_1'C$

$Y_2' = Y_3' - Y_1'C$

$Z_2' = Z_3' - Z_1'C$

$L_2^2 = X_2'^2 + Y_2'^2 + Z_2'^2$

The remaining unit vector **z** is a unit vector along B. Thus, from these unit vectors, one can derive the transformation matrix.

$$\begin{Bmatrix} x_i \\ y_i \\ z_i \end{Bmatrix} = \begin{bmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ t_{31} & t_{32} & t_{33} \end{bmatrix} \begin{Bmatrix} X_i \\ Y_i \\ Z_i \end{Bmatrix} \tag{2.3}$$

where subscript "i" pertains to the nodal point number and

$t_{11} = X_1'/L_1;\ t_{12} = Y_1'/L_1;\ t_{13} = Z_1'/L_1$

$t_{21} = Y_2'/L_2;\ t_{22} = Y_2'/L_2;\ t_{23} = Z_2'/L_2$

$t_{31} = (Y_1'Z_2' - Y_2'Z_1')/\ (L_1 L_2)$

$$t_{32} = (X_2'Z_1' - X_1'Z_2')/ (L_1 L_2)$$

$$t_{13} = (X_1'Y_2' - X_2'Y_1')/ (L_1 L_2)$$

In Figure 2., let area coordinates θ_i (i pertains to the nodal point number) be defined as

$$\theta_i = A_i/A = a_i + b_i x + c_i y \tag{2.4}$$

where $a_1 = (x_2y_3 - x_3y_2)/2A$; $b_1 = (y_2 - y_3)/2A$; $c_1 = (x_3 - x_2)/2A$

$a_2 = (x_3y_1 - x_1y_3)/2A$; $b_2 = (y_3 - y_1)/2A$; $c_2 = (x_1 - x_3)/2A$

$a_3 = (x_1y_2 - x_2y_1)/2A$; $b_3 = (y_1 - y_2)/2A$; $c_3 = (x_2 - x_1)/2A$

A being the element area.

The derivatives of the area coordinates with respect to the local coordinate system are given by

$$\frac{\partial\theta_i}{\partial x} = b_i = \frac{d_i}{2A} \; ; \qquad \frac{\partial\theta_i}{\partial y} = c_i = \frac{r_i}{2A} \tag{2.5}$$

where d_i and r_i are the projections of each side of the element on the x and y axes of the local coordinate system respectively and are given by

$$d_1 = t_{21}\, a_2 + t_{22}\, b_2 + t_{23}\, c_2 \tag{2.6}$$

$$d_2 = -(t_{21}\, a_1 + t_{22}\, b_1 + t_{23}\, c_1)$$

$$d_3 = -(d_1 + d_2)$$

$$r_1 = -(t_{11}\, a_2 + t_{12}\, b_2 + t_{13}\, c_2)$$

$$r_2 = t_{11}\, a_2 + t_{12}\, b_2 + t_{13}\, c_1$$

$$r_3 = -(r_1 + r_2)$$

where $a_1 = X_1 - X_3$
$a_2 = X_2 - X_3$
$b_1 = Y_1 - Y_3$
$b_2 = Y_2 - Y_3$
$c_1 = Z_1 - Z_3$
$c_2 = Z_2 - Z_3$

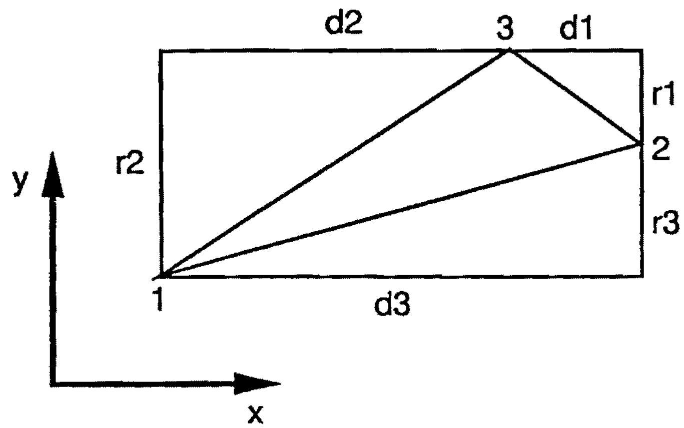


Figure 2. Geometrical representation of parameters d, r.

3.2 Modeling of the in-plane behavior

The local stiffness matrix of the CST element for the in-plane behavior can be shown to be given by[2]

$$[K] = h/4A\, [P]^t\, [C]\, [P] \qquad (2.7)$$

where h = thickness of element

$2A = d_i r_j - d_j r_i$

(d_i , r_i are the same as described in Eq.(2.6))

$$[P] = \begin{bmatrix} d_1 & d_2 & d_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & r_1 & r_2 & r_3 \\ r_1 & r_2 & r_3 & d_1 & d_2 & d_3 \end{bmatrix}$$

and [C] is the appropriate material coefficient matrix (for details see page 48)

Similarly, the local mass matrix of the CST element is calculated as

$$[M] = \frac{\rho h A}{12} \begin{bmatrix} 2 & 1 & 1 & 0 & 0 & 0 \\ 1 & 2 & 1 & 0 & 0 & 0 \\ 1 & 1 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 1 & 1 \\ 0 & 0 & 0 & 1 & 2 & 1 \\ 0 & 0 & 0 & 1 & 1 & 2 \end{bmatrix} \qquad (2.8)$$

where ρ = the mass density.

The work equivalent nodal vectors due to edge pressure loading is obtained as

$$\{R_p\} = [B_p]\, \{P\} \qquad (2.9)$$

with

$$[B_p] = \int_S [\Phi_b]^t [\Phi_p]\, dS$$

2 Details of stiffness matrix derivations can be found in reference [17]

$\{P\}$ = nodal values of pressure intensities

$[\Phi_b] = [\Phi_p] = [\theta_1, \theta_2, \theta_3]$ is the matrix of shape functions

Likewise, the work equivalent nodal vector due to thermal loading is given by

$$\{F_x\} = \frac{EhC_x}{6\mu} \left\{ \begin{array}{c} d_1T_1 \\ d_2T_2 \\ d_3T_3 \\ (d_1+d_2)T_1+(d_2+2d_1)T_2+(d_1+d_2)T_3 \\ (d_3+d_2)T_1+(d_2+2d_3)T_2+(d_3+2d_2)T_3 \\ (d_1+2d_3)T_1+(d_1+d_3)T_2+(2d_1+d_3)T_3 \end{array} \right\} \tag{2.10}$$

where d_i , r_i : same as described in Eq.(2.4)

$\mu = 1 - v^2$ (for plane-stress)

$1 - 2v$ (for plane-strain)

T_i = temperature differences, i = 1, 2, 3

C_x = thermal expansion coefficient in the x-direction

$\{F_y\}$ can be determined by replacing d_i ,C_x in Eq.(2.10) by r_i ,C_y with C_y being the thermal expansion coefficient in the y-direction.

3.3 Modeling of the plate bending behavior

As shown in Figure 3-e, the quadrilateral element (Q-12) with 12 degrees of freedom is used for modeling plate bending behavior.

This Q-12 element formulation is derived from the assemblage of four LCCT-9 (Linear Curvature Compatible Triangular elements, each with 9 degrees of freedom consisting of 3 degrees of freedom at each corner node (Figure 3-c)) followed by the elimination of the interior nodal degrees of freedom through static condensation.

The LCCT-9 element can be obtained from the LCCT-12 element as described below. The LCCT-12 element is in itself generated from an assemblage of three triangular subelements (Figure 3-b). After the assemblage of the three subtriangles, the degrees of freedom of the interior node are removed by static condensation leaving a fully conforming triangular element with 12 degrees of freedom.

Because mid-point nodes and the associated normal derivatives are somewhat cumbersome to handle from a computer programming consideration and because they tend to complicate mesh generation procedures and tend to increase bandwidth, it is desirable to eliminate the normal derivative degree of freedom along the edges of this triangle by requiring that the normal derivative along this edge vary linearly. This leads to the smooth change of the LCCT-12 element formulation to the LCCT-9 element formulation.

The formulation details of the LCCT-12 element are too complicated to be outlined in full in this report. We only present the bare minimum highlights of the formulation and the reader is directed to references [10] and [17] for all additional details.

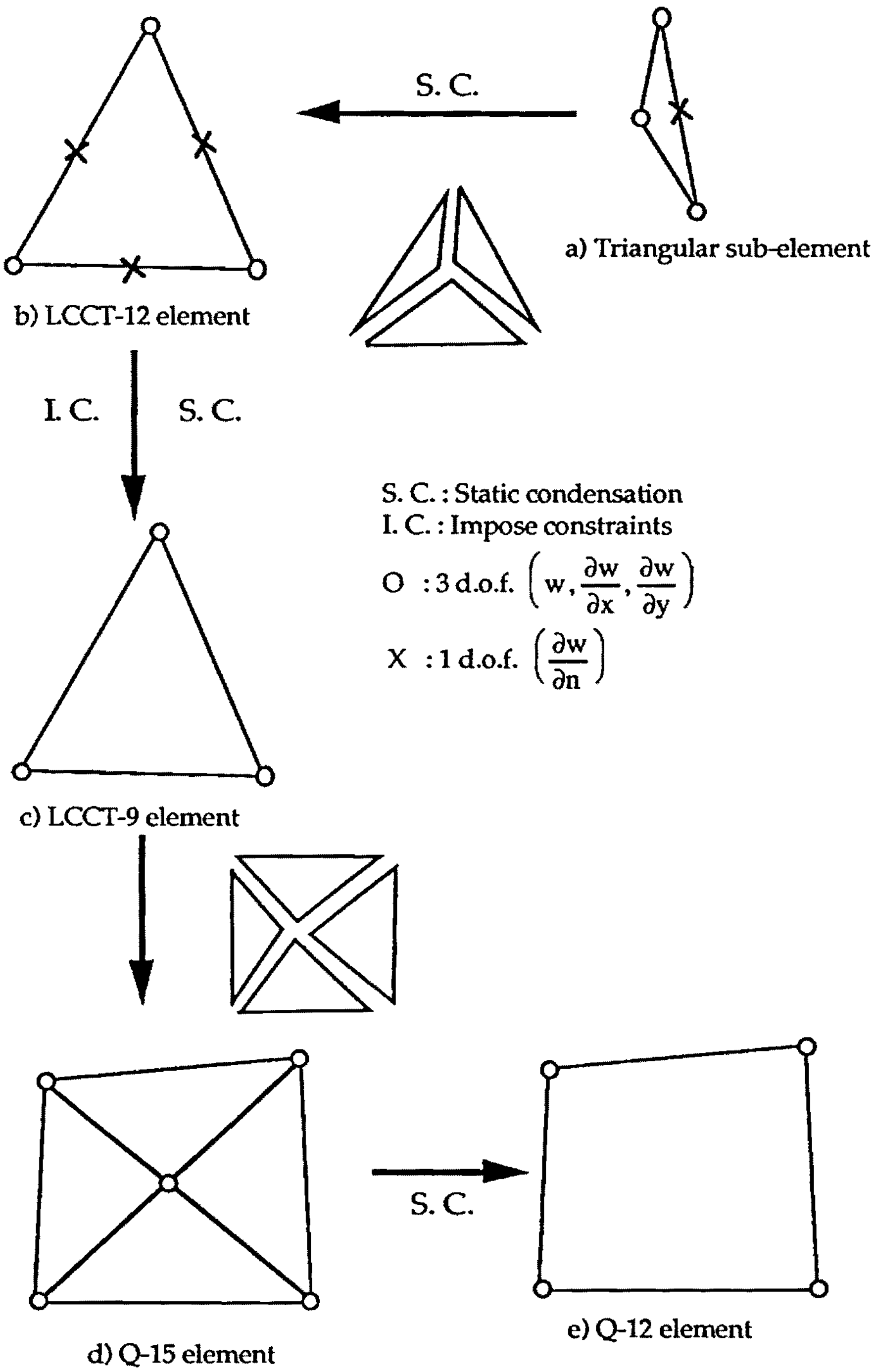


Figure 3. Generation of the Q-12 plate-bending element.

The local stiffness matrix of each of the subtriangular elements which make up the LCCT-9 element may be shown to be [10]

$$[K^{(i)}] = [T_n^{(i)}]^t[G^{(i)}]\,[T_n^{(i)}]$$

where "i" being the number of subtriangular element

$[T_n^{(i)}]$ being the matrix that relates element curvatures to the vector of element nodal degrees of freedom.[3]

$$[G^{(i)}] = \int_A \frac{h^3}{12}\{\theta\}^T[C]\{\theta\}\,dA$$

For a constant thickness element

$$[G] = \frac{h^3 A}{144}[C]\begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix}$$

For a variable thickness element

$$[G] = \frac{h_o^3 A}{432}[C][Q]$$

where [Q] is a 3x3 symmetric matrix given by

$$Q_{11} = (10\theta_1^3 + \theta_2^3 + 6\theta_3^3 + 6\theta_1^2\theta_2 + 6\theta_1^2\theta_3 +$$

$$3\theta_2^2\theta_1 + \theta_2^2\theta_3 + 3\theta_3\theta_1 + \theta_3^2\theta_2 + 3\theta_1\theta_2\theta_3)/17.5$$

$$Q_{12} = (4\theta_1^3 + 4\theta_2^3 + \theta_3^3 + 6\theta_1^2\theta_2 + 3\theta_1^2\theta_3 +$$

$$6\theta_2^2\theta_1 + 3\theta_2^2\theta_3 + 2\theta_3\theta_1 + 2\theta_3\theta_2 + 4\theta_1\theta_2\theta_3)/35$$

$$Q_{13} = (4\theta_1^3 + \theta_2^3 + 4\theta_3^3 + 3\theta_1^2\theta_2 + 6\theta_1^2\theta_3 +$$

---

3 Details of the formulation can be found in reference [17]

$$2\theta_2{}^2\theta_1 + 2\theta_2{}^2\theta_3 + 6\theta_3{}^2\theta_1 + 3\theta_3{}^2\theta_2 + 4\theta_1\theta_2\theta_3)/35$$

$$Q_{22} = (\theta_1{}^3 + 10\theta_2{}^3 + \theta_3{}^3 + 3\theta_1{}^2\theta_2 + \theta_1{}^2\theta_3 +$$

$$6\theta_2{}^2\theta_1 + 6\theta_2{}^2\theta_3 + \theta_3{}^2\theta_1 + 3\theta_3{}^2\theta_2 + 3\theta_1\theta_2\theta_3)/17.5$$

$$Q_{23} = (\theta_1{}^3 + 4\theta_2{}^3 + 4\theta_3{}^3 + 2\theta_1{}^2\theta_2 + 2\theta_1{}^2\theta_3 +$$

$$3\theta_2{}^2\theta_1 + 6\theta_2{}^2\theta_3 + 3\theta_3{}^2\theta_1 + 6\theta_3{}^2\theta_2 + 4\theta_1\theta_2\theta_3)/35$$

$$Q_{33} = (\theta_1{}^3 + \theta_2{}^3 + 10\theta_3{}^3 + \theta_1{}^2\theta_2 + 3\theta_1{}^2\theta_3 +$$

$$\theta_2{}^2\theta_1 + 3\theta_2{}^2\theta_3 + 6\theta_3{}^2\theta_1 + 6\theta_3{}^2\theta_2 + 3\theta_1\theta_2\theta_3)/17.5$$

and

$$h_o = \frac{1}{A}\int_A h(\theta_1, \theta_2, \theta_3)\, dA$$

with θ_i's being the area coordinates of the subtriangles.

Accordingly, the local stiffness matrix of the LCCT-9 element is given by

$$[K] = [K^{(1)}] + [K^{(2)}] + [K^{(3)}]$$

As shown in the fig. 3-d, four LCCT-9 elements consist of one Q-15 element. The stiffness matrix of the Q-12 element must then be established through static condensation.

The pressure load vector of the LCCT-9 element is calculated as

$$\{P\} = \{P^{(i)}\} \int_A \{\Phi^{(i)}\}\, \{\theta\}^t\, \{P_j\}\, dA$$

where $\{P^{(i)}\}$ = equivalent pressure loading for the subtriangular element i.

$\{P_j\}$ = pressure intensities at each nodal point of each subtriangular element.

$\{\theta\} = [\ \theta_1, \theta_2, \theta_3\ ]$ are area coordinates of the subtriangles

$\{\Phi^{(i)}\}$ is the matrix of shape functions for the LCCT-9 element. [4]

Similarly, the thermal load vector of the LCCT-9 element is calculated as

$$\{F_t\} = \{F_t^{(i)}\} = \alpha\beta h\,[T_n^{(i)}]^t \int_A [\theta]^t\,[D]\{\theta\}^t\,dA\,\{\theta_i\}$$

where i is the number of the subtriangular element
α is the thermal expansion coefficient
β is the temperature gradient across the thickness.

Finally, the local mass matrix of the LCCT-9 element for transverse deformations is obtained as

$$[M] = \rho \int_A h\{\Phi^{(i)}\}^t\{\theta\}^t\{\theta\}\{\Phi^{(i)}\}\,dA$$

where ρ = mass density and $\{\Phi^{(i)}\}$ as defined above.

4 See reference [10] for details.

4. Structure of the SAPNEW program

Figure 4. outlines the structure of the SAPNEW program.

Currently this program is developed with the objective of high speed execution under the in-core memory access environment. Therefore, this program uses the one-dimensional common array "A(5000000)" as the major storage array.

First, in the main program, the program reads the master control data for the analysis such as the title, the type of analysis, the number of nodal points, the number of load cases or highest vibration mode, the program execution mode, and control data for the eigen value/vector analysis.

Next, in the subroutine "INPUT", the program reads the nodal coordinates, the boundary conditions, the temperature of each nodal point.

In the case of static analysis, the subroutine "LOADS" reads the data pertaining to concentrated loads at given nodes and in given directions and constructs the load vector for the assembled model. The vector is then stored in memory.

In the case of dynamic analysis, the subroutine "MASSIN " reads the data pertaining to the non-structural concentrated masses and constructs the diagonal matrix as a vector and stores it into memory.

In the subroutines "ELCAL", "ELEMNT" and "SHELL", the program accesses the "TPLATE" routine which is the driver of the subroutines associated with the shell element.

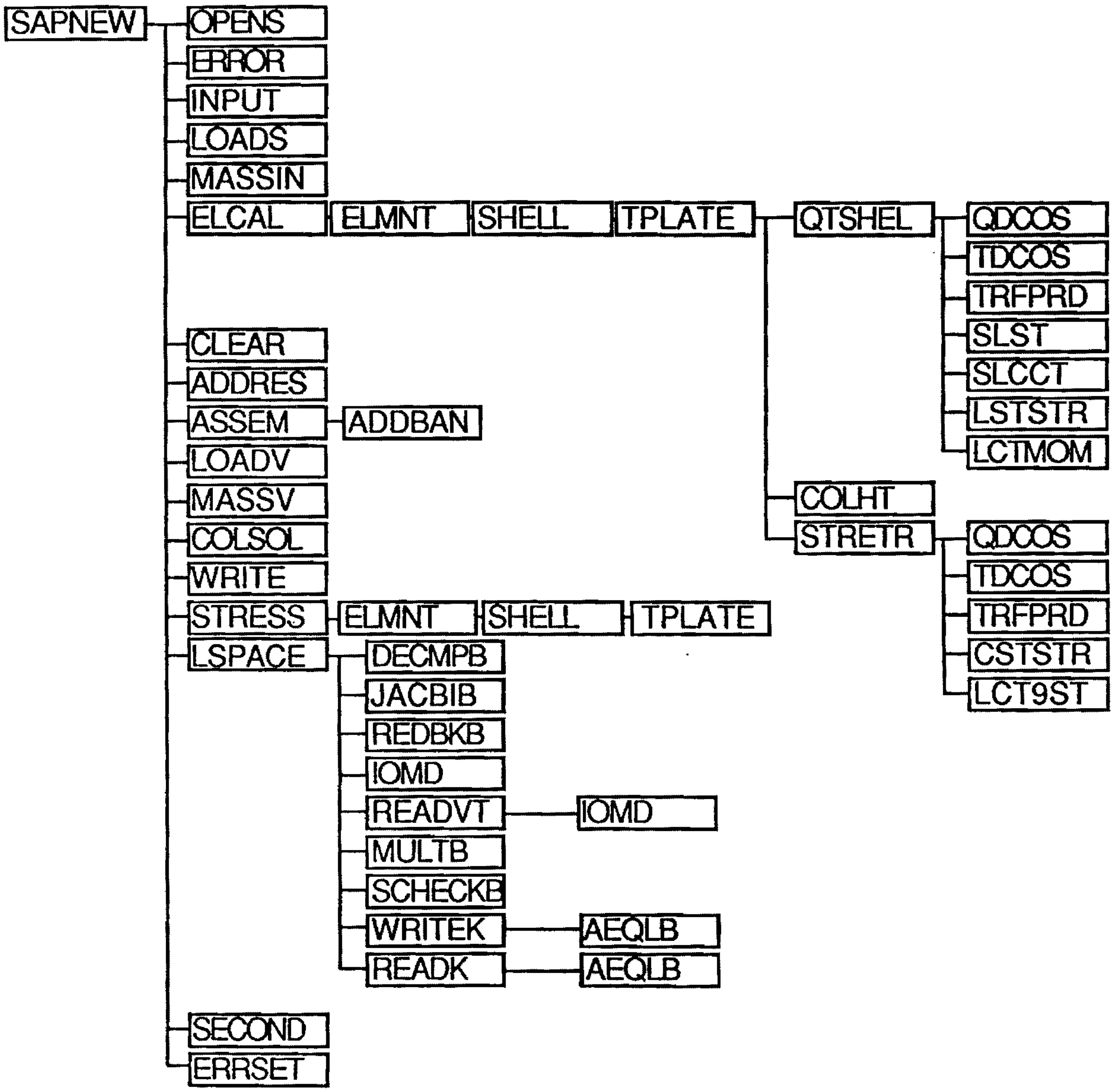


Figure 4. Structure of the SAPNEW program

In the subroutine "TPLATE", the program reads the material properties, the element load multiplication factor, element connectivity data, and calculates the stiffness matrix, the load vector and matrices required for the computations of element stresses at designated locations within the element.

These procedures are determined by the value of the variable "KKK" as follows;

KKK = -1 : Element stiffness matrix calculation
= 1 : Load vector calculation
= 2 : Matrices for stress calculation

In the subroutine "ADDRES", the sparsity structure of the assembled matrix is established. Subroutine "ASSEM" superimposes element stiffness matrices to obtain the global stiffness matrix.

In the case of the static analysis, the subroutine "LOADV" superimposes load vectors associated with pressure loading, thermal effects and self weight to obtain the assembled load vector.

In the subroutine "COLSOL", the global stiffness matrix of the system is decomposed into the $L\ D\ L^T$ form and displacements are calculated by forward and backward substitutions for each of the load cases.

Finally, stresses are calculated in the routine "STRESS" which accesses the routines "ELCAL", "ELEMNT", "SHELL", and "STRSC" in that order.

In the case of the dynamic analysis, eigen values and vectors are calculated by the subspace iteration method[5] in the subroutine "LSPACE". In this subroutine, the initial iteration vector is set-up and the subroutine "DECMPB" is called to transform the global stiffness matrix of the system into the L D L^T form for later usage. Subspace iteration is then performed by accessing routines "REDBKD", "MULTB" and "JACBIB". When the given error tolerance is achieved, the iteration procedure is ended and the Sturm sequence check is applied in the subroutine "SCHECKB".

As a special feature of this program, the restart option of the eigen value/vector analysis is available. This option uses the following characteristics of the subspace iteration method.

If some eigen values/vectors have already been calculated and more eigen values/vectors are needed, the previously calculated eigenvectors can be effectively utilized.

In the case of the restart option for the eigen value/vector analysis, the main program reads the master control data for the analysis such as the title, the type of analysis, the number of nodal points, the number of load cases or highest mode and the execution mode.

The main program then skips the input associated with the conventional nodal point input, the concentrated nodal masses input, and calculations associated with the element stiffness and mass matrices.

The main program then reads the ID-array information, the assembled stiffness and mass matrices, and the previously evaluated eigen vectors from a temporary file, and performs the eigen value/vector analysis by accessing the "LSPACE" subroutine.

---

[5] See reference [12]

5. Structure of the "TPLATE" subroutine for the shell element

The functions of the routine "TPLATE" are accomplished using two major subroutines; one is the "QTSHEL" routine which handles the calculation of the element stiffness matrix and the element stresses, and the other is the "STRETR" routine which generates the transformation matrix required for the calculation of stresses. This transformation is necessary as a result of the condensation procedure of the element stiffness matrix from the Q-19 element (19 D.O.F) to the Q-12 element (12 D.O.F).

a) QTSHEL routine

This routine consists of seven subroutines, namely; QDCOS, TDCOS, TRFPRD, SLST, SLCCT, LSTSTR, LCTMOM.

The direction cosines of the local coordinate system associated with the element are calculated in the "QDCOS" routine, and the area coordinates of the sub-triangular elements stemming from the division of the shell element, are calculated in the "TDCOS" routine.

In the "TRFPRD" routine, the transformation matrix associated with the transformation of the sub-triangular element to the macro triangular element is calculated.

The subroutine "SLST" calculates the stiffness matrix representing the plate stretching effect of the shell element. The stiffness matrix, which represents the plate bending effect, is calculated in the routine "SLCCT".

In the case of thermal loading, the in-plane loads and the bending moments are calculated in the two routines, "LSTSTR" and "LCTMOM".

b) STRETR routine

This routine consists of five subroutines, namely; QDCOS, TDCOS, TRFPRD, CSTSTR, LCT9ST.

The functions of three subroutines (QDCOS, TDCOS, TRFPRD) are the same as the functions outlined in the "QTSHEL" routine. The transformation matrix for the stresses due to in-plane loads is calculated in the subroutine CSTSTR, while the LCT9ST routine calculates the transformation matrix required for the calculation of stresses due to bending moments.

6. Comparison of results - Effectiveness of vectorization / parallelization

To check the accuracy of the SAPNEW program and evaluate its effectiveness in a vector/concurrent processing environment, several static and dynamic analyses of rectangular plates were carried out for various aspect ratios, and mesh-sizes.

Descriptions of models are listed in Table 1. The results of the static analysis are listed in Table 2. The results of the dynamic analysis are listed in Table 3. Finally, the results of the dynamic restart analysis are listed in Table 4.

Table 1. Description of models

Model no	1	2	3	4	5	6	7	8
Aspect ratio (b/a)	1.0	1.0	1.0	1.0	1.4	1.4	1.4	1.4
Mesh size	10x10	20x20	30x30	50x50	10x10	20x20	30x30	50x50
Total D.O.F	287	1167	2649	7409	287	1167	2649	7409
Mean bandwidth	30	61	96	156	30	61	96	156

Notes:

boundary condition : simple supports on all four sides
plate length : a = 20.0 m
bending rigidity : 0.8333e-1 N-m
mass density : 1.0e-4 kg (mass)
loading type
- Concentrated load applied at mid-point of plate. (F = 1.0 N)
- Uniform pressure load (p = 0.1 N/m2)

Table 2. The results of static analysis

Aspect ratio of shell element	Loading type	Mesh size	Maximum deflection (mm)	theory (mm)	relative error(%)	speed-up ratio
1.0	F	10x10	55.007	55.903	1.60	4.82
		20x20	55.484		0.74	4.94
		30x30	55.623		0.50	5.58
		50x50	55.847		0.10	8.01
	p	10x10	764.31	782.65	2.34	4.91
		20x20	776.04		0.84	4.93
		30x30	779.51		0.41	5.62
		50x50	781.08		0.11	7.86
1.4	F	10x10	70.329	71.518	1.66	4.92
		20x20	71.050		0.65	4.98
		30x30	71.303		0.31	5.60
		50x50	71.374		0.20	8.12
	p	10x10	1333.4	1359.04	1.88	4.87
		20x20	1353.5		0.41	5.01
		30x30	1361.1		0.15	5.80
		50x50	1358.9		0.10	8.09

Notes:

F : concentrated load at the mid-point of plate
p : uniform pressure load
speed-up ratio : execution time of sequential code is divided by the execution time of vectorization & parallelization code

Table 3. The results of the dynamic analysis

Model no.		Frequencies of modes (Hz)							speed-up ratio
		1	2	3	4	5	6	7	
1	C	4.5717	11.331	11.331	18.216	22.776	22.776	29.777	8.0
	T	4.5048	11.262	11.262	18.019	22.524	22.524	29.281	
	E	1.5	0.6	0.6	1.1	1.1	1.1	1.7	
2	C	4.5079	11.279	11.279	18.069	22.587	22.587	29.406	8.0
	T	4.5048	11.262	11.262	18.019	22.524	22.524	29.281	
	E	0.06	0.15	0.15	0.28	0.27	0.27	0.4	
3	C	4.5061	11.269	11.269	18.041	22.551	22.551	29.336	10.42
	T	4.5048	11.262	11.262	18.019	22.524	22.524	29.281	
	E	0.02	0.06	0.06	0.12	0.1	0.1	0.18	
4	C	4.5053	11.264	11.264	18.027	22.534	22.534	29.301	12.89
	T	4.5048	11.262	11.262	18.019	22.524	22.524	29.281	
	E	0.01	0.02	0.02	0.04	0.04	0.04	0.68	
5	C	3.4594	6.9313	10.291	13.208	19.564	20.845	27.752	8.0
	T	3.4016	6.8492	10.159	13.065	19.352	20.639	27.396	
	E	1.7	1.2	1.3	1.1	1.1	1.0	1.3	
6	C	3.4458	6.9176	10.230	13.143	19.352	20.701	27.451	8.0
	T	3.4016	6.8492	10.159	13.065	19.352	20.639	27.396	
	E	1.3	1.0	0.7	0.6	0.8	0.3	0.2	
7	C	3.4390	6.9245	10.230	13.104	19.448	20.680	27.451	10.43
	T	3.4016	6.8492	10.159	13.065	19.352	20.639	27.396	
	E	1.1	1.1	0.7	0.3	0.5	0.2	0.2	
8	C	3.4322	6.8971	10.169	13.130	19.390	20.680	27.478	14.87
	T	3.4016	6.8492	10.159	13.065	19.352	20.639	27.396	
	E	0.9	0.7	0.1	0.5	0.2	0.2	0.3	

Notes: model no. : refer to Table 1.
C : calculated value
T : theoretical value[6]
E : relative error (%)
speed-up ratio : see the definition of Table 2.

Table 4. The comparison of the execution times for the restart option in the eigen value/vector analysis

Mesh size	Execution time (sec)			
	sequential.	vect.†	vect. & parallel (initial execution)	vect. & parallel (restart)
20x20	247.48	60.58	30.97	12.35
30x30	861.25	170.32	82.60	31.26
50x50	4952.12	690.15	322.62	173.31

Notes: Total modes : 7

modal information of restart : previously evaluated 5 modes

† Vectorization with global optimization.

7. Alternate solution procedure: The conjugate gradient algorithm

Current popular algorithms for solving a system of linear equations, such as the $[L][D][L]^T$ factorization scheme, were designed to operate in a sequential mode. Generally, however, these methods fail to take advantage of a parallel processing environment. Iterative solution schemes appear to have a greater potential to exploit multi-processing systems. The conjugate gradient method, as proposed by K. Law [14] , was implemented because of its ability to operate on an element-by-element basis and its good rate of convergence. The element-by-element preconditioned conjugate gradient method with a simple diagonal preconditioner appears to be ideal for a parallel computing environment and hence the reason for its implementation into SAPNEW.

The conjugate gradient algorithm

The conjugate gradient algorithm for the solution of the linear system,

$$[K]\{q\} = \{F\},$$

can be given as:

$$\text{Select } \{q\}_o$$
$$\{r\}_o = \{F\} - [K]\{q\}_o$$
$$\gamma_o = \{r\}^t_o \{r\}_o$$
$$\{p\}_o = \{r\}_o$$

Begin Loop: i=1

$$\{u\}_i = [K]\{p\}_{i-1}$$
$$\alpha_i = \gamma_{i-1} / \{p\}^t_{i-1} \{u\}_i$$
$$\{q\}_i = \{q\}_{i-1} + \alpha_i \{p\}_{i-1}$$
$$\{r\}_i = \{r\}_{i-1} - \alpha_i \{u\}_i$$
$$\gamma_i = \{r\}^t_i \{r\}_i$$
$$\beta_i = \gamma_i / \gamma_{i-1}$$
$$\{p\}_i = \{r\}_i + \beta_i \{p\}_{i-1}$$

i --> i+1 : Repeat Loop

The vector {r} is the negative of the gradient of the total potential energy function and is equal to the force residual of the system of equations. The vectors $\{p\}_i$ are the [K]-conjugate search directions. The algorithm terminates after the m^{th} iteration if the magnitude of $\{r\}_m$ is small compared to the magnitude of the applied load vector {F}. A reasonable first guess for the initial displacement vector $\{q\}_o$ is made by assuming that the stiffness matrix [K] can be approximated by a diagonal matrix. If [d] is the matrix of the diagonal elements of [K] then: $\{q\}_o = [d]^{-1} \{F\}$. This also stems from the diagonal preconditioner that is applied to the matrix [K] in order to accelerate convergence to the solution.

Multiple load cases

Information about the system generated during the iterative process (namely the conjugate directions $\{p\}_i$) can be saved and used in subsequent load cases to provide a better first guess for {q}. If [P] is the matrix whose columns are the conjugate directions $\{p\}_i$ then let $[P]^t [K] [P] = [D]$ (a diagonal matrix). Hence, $\{q\}_o = [P] [D]^{-1} [P]^t \{F\}$. Note that if all of the $\{p\}_i$ directions have been calculated then $[P] [D]^{-1} [P]^t = [K]^{-1}$ exactly. Hence, even with a few $\{p\}_i$ directions, $\{q\}_o = [P] [D]^{-1} [P]^t \{F\}$ may still provide a good initial guess and the number of iterations required for convergence may be only a small fraction of the total degrees of freedom.

Preconditioning

To improve convergence of the iterative scheme, a diagonal preconditioner [T] can be applied to the original system, $[K] \{q\} = \{F\}$, such that [15]

$$[T]^t [K] [T] [T]^{-1} \{q\} = [T]^t \{F\}$$

becomes

$$[k] \{y\} = \{f\}$$

where

$$[k] = [T]^t [K] [T]$$
$$\{y\} = [T]^{-1} \{q\}$$
$$\{f\} = [T]^t \{F\}$$

The preconditioner [T] is selected such that the diagonal elements of [k] are all equal to one. This transformation increases the rate of convergence of the algorithm. Once the vector, {y}, is determined by the conjugate gradient method, the vector, {q}, is calculated with the transformation

$$\{q\} = [T] \{y\}.$$

Implementations

There are four different types of operations involved in the conjugate gradient algorithm. These are scalar division, vector sums and dot products and matrix-vector multiplication. The vector operations are easily handled by vector processors. However, due to the compact skyline storage scheme for the stiffness matrix, the calculation of the product [K] {p} = {u} cannot be fully vectorized. This is the primary limitation in the implementation of the conjugate gradient algorithm on the global level.

The element-by-element implementation stores each element's stiffness matrix $[K]^e$. Then the products $[K]^e \{p\}^e = \{u\}^e$ for each element is calculated simultaneously by different processors in a multi-processing system. This parallelism is of the type that should be readily supported by the single instruction-multiple data processing environment provided by the Alliant FX/Fortran compiler. The vector {u} is assembled by summing appropriate components of $\{u\}^e$ over the elements.

Subroutines

Two subroutine packages using the conjugate gradient algorithm have been written to solve the linear systems for the SAPNEW finite element program. Each set is linked with the object code SAP.O to generate an executable version of SAPNEW using that alternate solution scheme.

Element-by-element

The program named SFECGM.F contains seven subprograms to implement the element-by element version of the conjugate gradient algorithm. It contains slightly modified versions of the main program and the subroutine TPLATE from SAPNEW. The subroutine PRECON formulates and applies the diagonal preconditioner to the element stiffness matrices. The subroutine POSTCN "unconditions" the solution vector to generate the solution to the original system. The subroutine MAKKE performs initializing tasks for the main conjugate gradient subroutine. The subroutine ECGM contains the set-up and main iteration loop of the conjugate gradient algorithm. The susubroutine KP performs the element-by-element matrix-vector multiplication.

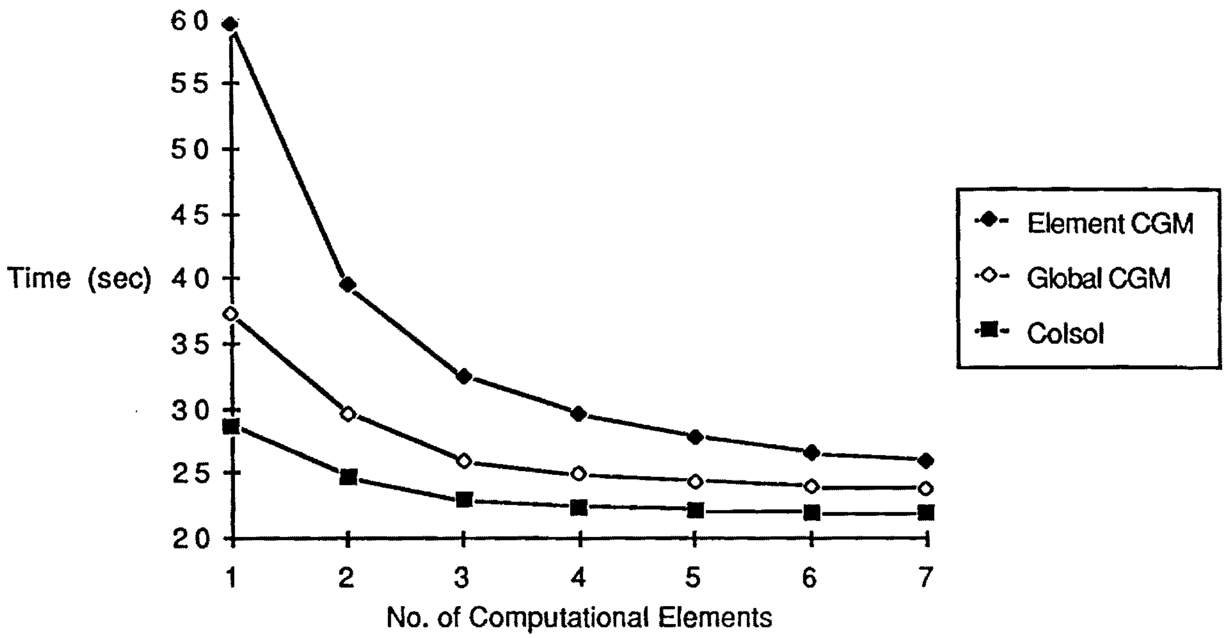
Global

The program named SFCGM.F contains six subprograms to implement the conjugate gradient algorithm on the global system of equations. It contains the subroutine TPLATE and a slightly modified version of the main program from SAPNEW as well as the subroutine MULT [12]. The subroutines named PRECON and POSTCN implement the diagonal preconditioner in the same manner as in the element-by-element version. The subroutine CGM contains the set-up and main iteration loop of the conjugate gradient algorithm.

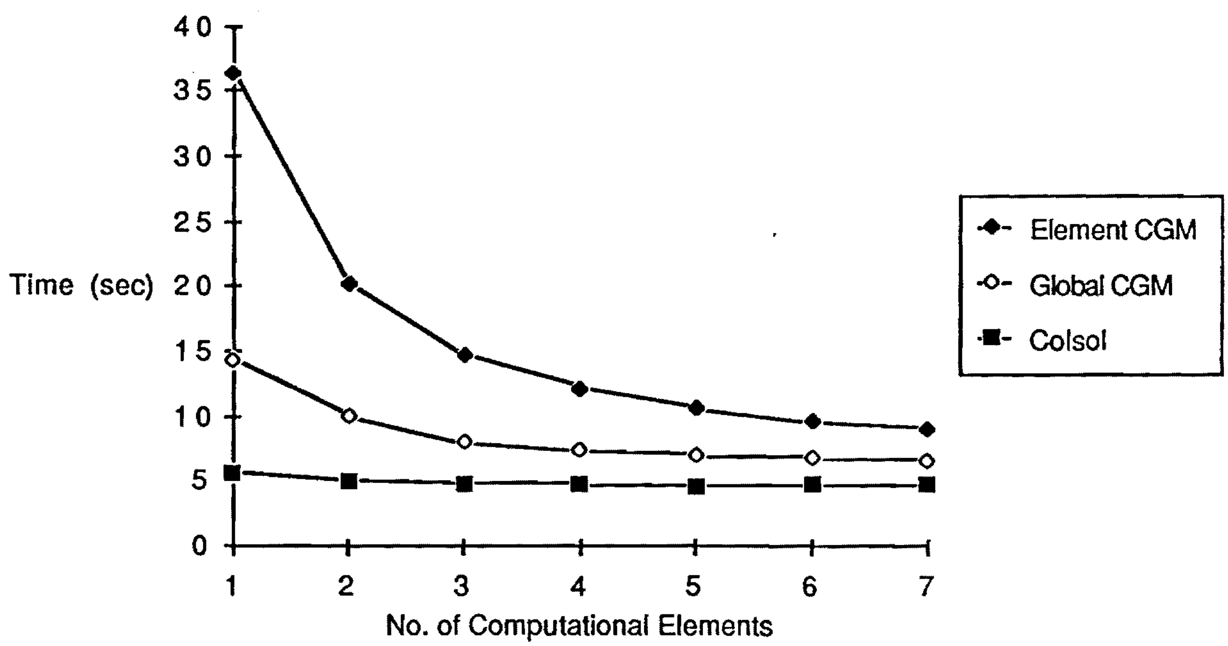
Testing and results

A test model containing 400 elements (20x20 mesh - see Table 1.) and 1159 degrees of freedom was generated. The static solution for a single central concentrated load was computed with the element-by-element (ECGM) and the global (CGM) versions of the conjugate gradient algorithm as well as the $[L][D][L]^T$ factorization scheme (the subroutine COLSOL), each running on one to seven processors. The solution times required show that these implementations of the conjugate gradient method are not yet cost effective (see Figure 5). However, the speed-ups (defined as the time required for one processor with vectorization and global optimization divided by the time required for n-processors with vectorization and global optimization) for the various methods show that: 1. COLSOL receives almost no benefit from parallelization. 2. The global conjugate gradient method receives only a moderate benefit from parallelization. 3. The element-by-element conjugate gradient method's performance is greatly enhanced when additional processors are used (see Figure 6).

These results suggest that although additional efforts are needed to make this method cost effective, the inherent parallelism of the element-by-element conjugate gradient algorithm should make it the method of choice for parallel processing machines, especially the concurrent machines of the future which can be expected to have large numbers of processors.

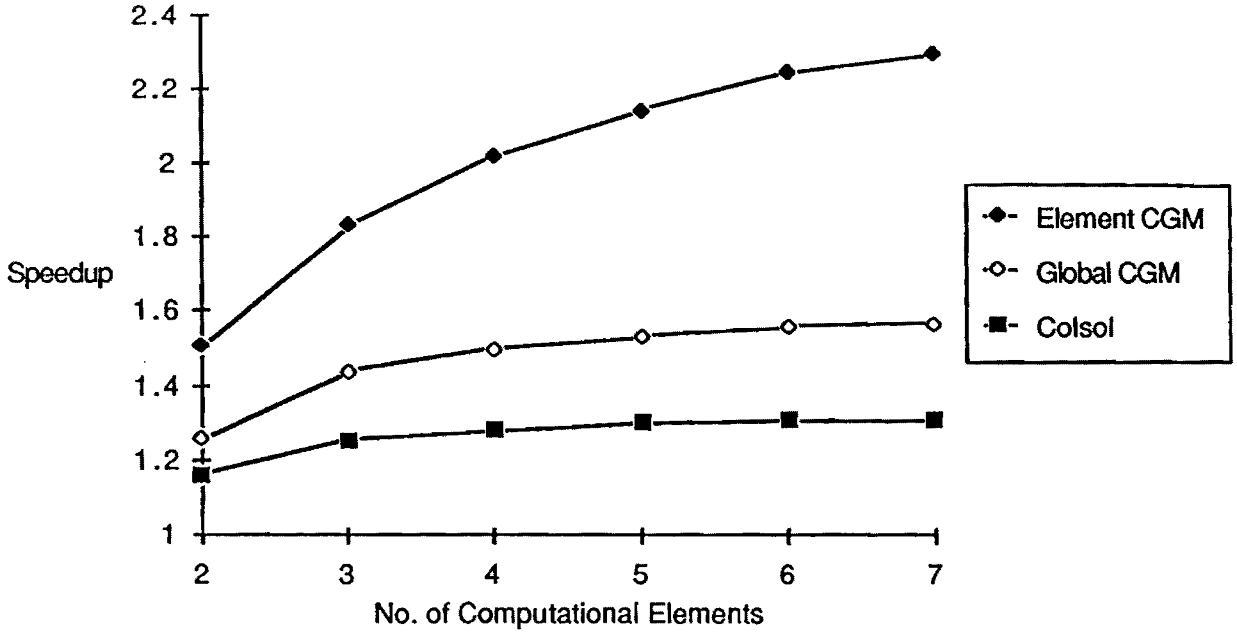


(a)

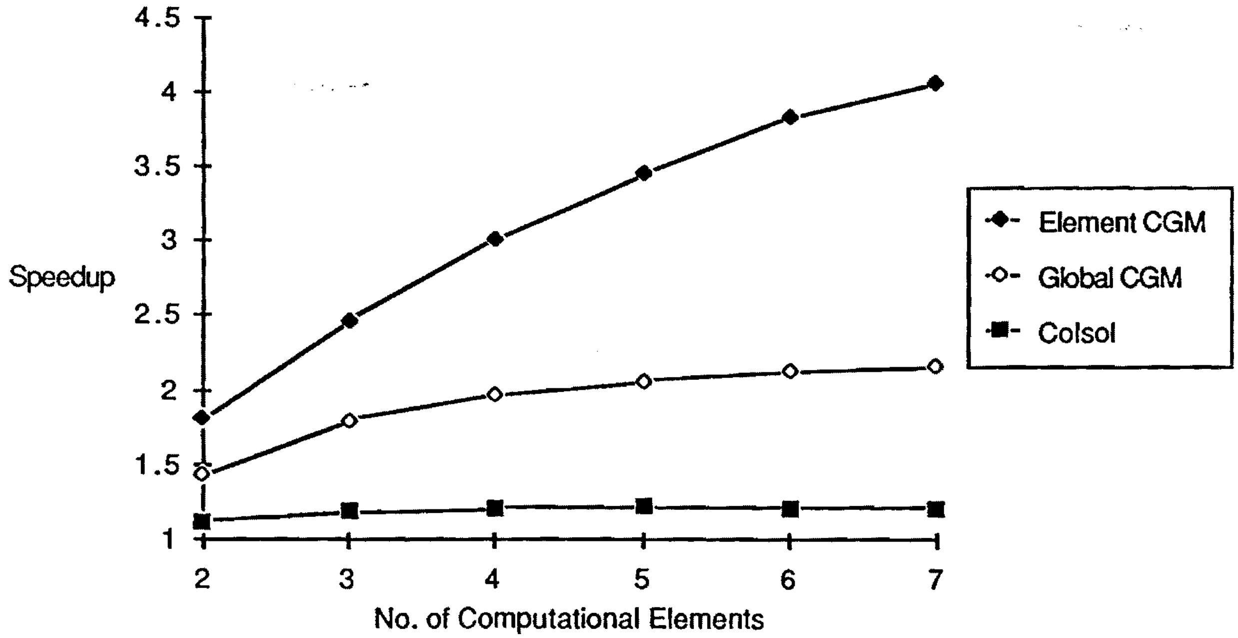


(b)

Figure 5. Time required for 20x20 mesh problem solution; (a) Total, (b) Linear system solution phase.



(a)



(b)

Figure 6. Speedups for 20x20 mesh problem solution; (a) Total, (b) Linear system solution phase.

8. Summary

Work is continuing on the element-by-element conjugate gradient algorithm with a view to make it cost-effective on single and multiple load case situations. Work has also been initiated on applying the conjugate gradient method to the problem of finding eigenvalues and eigenvectors of the finite element model in a manner described by Fox and Kapoor [16].

In summary, the authors of this report believe that they have put together a state-of-the-art code for static and vibration analysis of thin shell type structures using a fully conforming shell element. For static analysis, the analyst has the option of using the LDL^T factorization scheme followed by forward and backward substitution for solving the linear system of equations which are stored using a compacted skyline storage scheme. Alternatively, an element-by-element or a global preconditioned conjugate gradient method for the solution of the linear system of equations is also available. For vibration analysis, the highly efficient subspace iteration technique for extracting eigenvalues and eigenvectors of interest is available, although plans exist for extending the preconditioned conjugate gradient method for this analysis as well. Material of the shell can be either isotropic or orthotropic and, in addition to the conventional mechanical loading, thermal loading with through the thickness gradients is permitted.

Even though, currently, the code does not exploit the capabilities of the Alliant FX/8 machine fully, it is hoped that, with an improved understanding of the machine architecture, its directives, etc., a highly efficient version of the SAPNEW program can be produced that can effectively address NASA's needs for the analysis of turbo machinery bladed disk assemblies.

References

[1] Kaza, K.R.V. and Kielb, R.E. , " Flutter and Response of Mistuned Cascade in Incompressible Flow", *AIAA Journal*, vol. **20**, no.2, pp.1120-1127

[2] Ayers, C.L. , "Concurrent Processing of Finite Element Calculations", M.S. thesis, Georgia Institute of Technology, July 1989.

[3] Hughes, T.J.R., et al. , "New Alternating Direction procedures in Finite Element Analysis based upon eBe Approximate Factorizations", *Computer Methods for Nonlinear solids and Structural Mechanics*, AMD vol.**54**, 1983,pp.75-109.

[4] Bathe , K.J., "ADINA - A Finite Element Program for Automatic Dynamic Incremental Nonlinear Analysis", Report 82448-1, Acoustics and Vibration Laboratory, MIT, Cambridge, MA,1975

[5] FX/FORTRAN Language manual, ALLIANT COMPUTER SYSTEMS Company, v.4.0,1988

[6] Concentrix user's guide, ALLIANT COMPUTER SYSTEMS Company, v.1.0,1988

[7] D. Parkinson ," The solution of N linear equations using P Processors", *Parallel computing* **83**, pp.81-113, Elsevier Science Publishers ,1983

[8] Olaf Storaasli, et. al. ," Solution of structural analysis problems on a parallel computer "

[9] Helen M. Doerr, et. al ," Improving vector performance", Cornell Super Computer facility, Cornell University, 1987

[10] Clough R.W. and Felippa, C.A.," A Refined Quadrilateral element for analysis of plate bending", *Proceedings 2'nd conference on matrix methods in structural mechanics*, Wright Peterson AFB, Ohio, 1968.

[11] Timoshenko S.P., Woinowsky-Krieger S. ,*Theory of plates and shells*, pp.140-158, McGraw-Hill company, 1959

[12] Bathe K.J., *Finite element procedures in engineering analysis* pp.672-684, Prentice-Hall company, 1982

[13] Leissa A.W., *Vibration of plates*, pp.44-45, NASA-SP160, 1969

[14] Kincho, H. Law ," Parallel Finite Element Solution Method", *Computer & Structures* vol.23, No.6, pp845-858,1986

[15] Argyris, J. et al., "Τα Παντα Ρει", *Computer Methods in Applied Mechanics and Engineering*, vol.51, pp.289-365,1985

[16] Fox, R.L. and Kapoor, M.P., *A minimization Method for the solution of the Eigenproblem arising in structural dynamics*, AFFDL-TR-68-150, pp.271-306.

[17] Felippa C.A., Refined finite element analysis of linear and Nonlinear two dimensional structures, SESM report 66-2, Dept. of civil eng., University of California, Berkeley,1966.

APPENDIX 1

USERS' GUIDE FOR THE SAPNEW PROGRAM

File names

<u>Main files</u>

1) linear equation solution scheme: LDL^T

Source code files : sapnew.f (static & dynamic capabilities)
static.f (only static capability)

Execution files : sapnew (static & dynamic capabilities)
static (only static capability)

2) Alternate linear equation solution scheme (only static capability)

Source code files : sap.f (main code)
sfcgm.f (global conjugate gradient solution routine)
sfecgm.f (element by element conjugate gradient solution routine)

Execution files : cgm.opt (global conjugate gradient solution)
ecgm.opt (element by element conjugate gradient solution)

Input file : **.dat
Output file : **.out

Note: 1. "**" means the user-defined file name.
2. In the case of static analysis, use " static " since it is faster than " sapnew ".

<u>Auxiliary files</u>

file for storage of modal information : modal.inf
file for storage of assembled stiffness matrix : stif.inf
file for storage of assembled mass matrix and the LM array : mass.inf

Note: These auxiliary files are necessary for restarting the dynamic analysis.

Card input form

Case 1. Static analysis

Heading card	(section I)
Nodal points information card	(section III)
Concentrated load card	(section IV)
Elements card	(section V)
Unit conversion factor card	(section VI)
Two blank cards	

Case 2. Eigen value/vector analysis

Heading card	(section I)
Control card for dynamic analysis	(section II)
Nodal points information card	(section III)
Concentrated nodal mass card	(section IV)
Elements card	(section V)
Two blank cards	

Case 3. Restarting the Eigen value/vector analysis

Heading card	(section I)
Control card for dynamic analysis	(section II)
Two blank cards	

I. Heading card (18A4,5I5)

I.A Heading card for the "sapnew" program (18A4,5I5)

variable	format	description
HED	18A4	title of analysis
NDYN	I5	analysis code NDYN = 0 ; static analysis NDYN = 1 ; eigen value/vector analysis
NUMNP	I5	number of total nodal points
NUMEG	I5	number of element groups
NLCASE	I5	number of load cases or modes NDYN = 0 ; NLCASE = no. of load cases NDYN = 1 ; NLCASE = no. of higest mode
MODEX	I5	index for the execution mode MODEX = 1 ; data check MODEX = 0 ; execution

I.B Heading card for the "static" program (18A4,4I5)

variable	format	description
HED	18A4	title of analysis
NUMNP	I5	number of total nodal points
NUMEG	I5	number of element groups
NLCASE	I5	number of load cases
MODEX	I5	index for the execution mode MODEX = 1 ; data check MODEX = 0 ; execution

NOTE: If NDYN = 0 (i.e. STATIC ANALYSIS), skip the control card for dynamic analysis (II) and refer to the nodal points input card section (III).

II. Control card for dynamic analysis (2F10.0,2I5,F10.0,4I5)

variable	format	description
COFQ	F10.0	cut-off frequency
RTOL	F10.0	error tolerance in the subspace iteration procedure
NITEM	I5	maximum no. of iterations
ISHIFT	I5	shifting code ISHIFT = 0 ; no shifting ISHIFT = 1 ; shifting
SHIFT	F10.0	shifting factor
IFSS	I5	flag for sturm sequence check IFSS = 0 ; no check IFSS = 1 ; check
IFPR	I5	flag for printing the iteration procedure IFPR = 0 ; do not print IFPR = 1 ; print
IGIVEN	I5	flag for restart execution IGIVEN = 0 ; initial execution IGIVEN = -1 ; restart
ISAVE	I5	flag for saving modal parameters ISAVE = 0 ; do not save ISAVE = 1 ; save for the later usage

Note: If more frequencies are needed, the value of variable "ISAVE" in the initial data file should be set to "1".

III. Nodal points information card (7I5,3F10.0,I5,F10.0)

variable	format	description
N	I5	node number
ID(6)	6I5	boundary condition code for the X, Y, Z, RX, RY, RZ direction ID = 0 ; free ID = 1 ; fixed
X(N)	F10.0	x-coordinate
Y(N)	F10.0	y-coordinate
Z(N)	F10.0	z-coordinate
KN	I5	node generation code. It should be described in the last node to be generated

Note: If "N" is equal to the number of total nodal points, the input procedure of this card will be terminated.

NOTE: If NDYN = 1 (i.e. EIGEN VALUE/VECTOR ANALSYS), skip the concentrated load card (IV.A) section and refer to the nodal mass card (IV.B) section.

IV. A Concentrated load card

Note: The following two cards (IV.A-1 & IV.A-2) should be repeated as many times as the total number of load cases (" NLOAD ")

IV. A-1 Load case card (2I5)

variable	format	description
LL	I5	number of the load case
NLOAD	I5	number of loads in the load case "LL"

IV. A-2 Description card for nodal load (2I5,F10.0)

variable	format	description
NOD	I5	node number at which the load is applied
IDIR	I5	code for the direction of the applied load IDIR = 1 x- direction = 2 y- direction = 3 z- direction = 4 rx-direction = 5 ry-direction = 6 rz-direction
FLOAD	F10.0	manitude of applied load

NOTE: If NDYN = 0 (i.e. STATIC ANALYSIS), skip the nodal mass card (IV.B) section and refer to the element card section (V)

IV. B Nodal mass card (I5,6F10.0)

variable	format	description
NOD	I5	node number at which lumped mass is specified.
XMASS	F10.0	nodal mass in the x-dir.
YMASS	F10.0	nodal mass in the y-dir.
ZMASS	F10.0	nodal mass in the z-dir.
RXMAS	F10.0	nodal inertia in the rx-dir.
RYMAS	F10.0	nodal inertia in the ry-dir.
RZMAS	F10.0	nodal inertia in the rz-dir.

Note: If " NOD " is zero, the input procedure of this card will be terminated.

V. Element cards

V-1. Element control card (3I5)

variable	format	description
NPAR(1)	I5	index for the element type
		NPAR(1) =1 ; shell element (currently only available element)
NPAR(2)	I5	number of elements
NPAR(3)	I5	number of material property groups

V-2. Material property card (I5,20X,F10.0,3F10.0/6F10.0)

variable	format	description
K	I5	material property number
DEN	F10.0	mass density
A(1,K)	F10.0	thermal expansion coef. in the x-dir.
A(2,K)	F10.0	thermal expansion coef. in the y-dir.
A(3,K)	F10.0	thermal expansion coef. in the z-dir.
C(1,K)	F10.0	C_{11} of the material coef. matrix $[C_{ij}]$
C(2,K)	F10.0	C_{12} of the material coef. matrix $[C_{ij}]$
C(3,K)	F10.0	C_{13} of the material coef. matrix $[C_{ij}]$
C(4,K)	F10.0	C_{22} of the material coef. matrix $[C_{ij}]$
C(5,K)	F10.0	C_{23} of the material coef. matrix $[C_{ij}]$
C(6,K)	F10.0	C_{33} of the material coef. matrix $[C_{ij}]$

Note: The material coefficient matrix $[C_{ij}]$ should be input as follows:

For isotropic materials:

Plane stress:

$$[C_{ij}] = \frac{E}{1-\nu^2}\begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix}$$

Plane strain:

$$[C_{ij}] = \frac{E}{(1+\nu)(1-2\nu)}\begin{bmatrix} 1-\nu & \nu & 0 \\ \nu & 1-\nu & 0 \\ 0 & 0 & \frac{1-2\nu}{2} \end{bmatrix}$$

For orthotropic materials:

Plane stress:

$$[C_{ij}] = \frac{E_y}{\left(1-n\nu_y^2\right)}\begin{bmatrix} n & n\nu_y & 0 \\ n\nu_y & 1 & 0 \\ 0 & 0 & m\left(1-\nu_y^2\right) \end{bmatrix}$$

Plane strain:

$$[C_{ij}] = \frac{E_y}{\left(1+n\nu_y\right)\left(1-2n\nu_y\right)}\begin{bmatrix} 1-n\nu_y & n\nu_y & 0 \\ n\nu_y & 1-n\nu_y & 0 \\ 0 & 0 & \frac{m\left(1-n\nu_y\right)}{2} \end{bmatrix}$$

where
- E : Young's modulus
- G : shear modulus
- ν : Poisson's ratio
- $n : E_x / E_y$
- $m : G_x / G_y$

When we determine the values of the directional material coefficients (E_x, E_y, etc.) for the stiffened plate, we should consider the effective breadth of the stiffener.

V-3. Load multiplier cards

V-3-1. Load multiplier card for pressure loading (4F10.0)

variable	format	description
PF(1)=>PF(4)	4F10.0	pressure load multiplier factors for load cases (1, 2, 3, 4)

V-3-2. Load multiplier card for thermal loading (4F10.0)

variable	format	description
PT(1)=>PT(4)	4F10.0	thermal load multiplier factors for load cases (1, 2, 3, 4)

V-3-3. Load multiplier cards for self weight

a) Cards for the x-direction acceleration (4F10.0)

variable	format	description
PWXF(1)=>PWXF(4)	4F10.0	thermal load multiplier factors for load cases (1, 2, 3, 4)

b) Cards for the y-direction acceleration (4F10.0)

variable	format	description
PWYF(1)=>PWYF(4)	4F10.0	thermal load multiplier factors for load cases (1, 2, 3, 4)

c) Cards for the z-direction acceleration (4F10.0)

variable	format	description
PWZF(1)=>PWZF(4)	4F10.0	thermal load multiplier factors for load cases (1, 2, 3, 4)

V-4. Element description card (7I5,4F10.0)

variable	format	description
NEL	I5	element number
NI	I5	first node number of the element
NJ	I5	second node number of the element
NK	I5	third node number of the element
NL	I5	fourth node number of the element
NO	I5	mid-point node number of the element
MTYP	I5	material property number
THICK	F10.0	thickness of the element
PR	F10.0	pressure on the element
TEMP	F10.0	temperature of the element
ZTEMPGRAD	F10.0	temperature gradient accross the thickness of the element

Notes:

1. If the element is triangular, the fourth node number and the mid-point node number should be set to zero.

2. If the element is quadrilateral and the behavior at the mid-point needs to be known, the mid-point node number should be descibed.

VI. Unit conversion factors for the load (4F10.0)

variable	format	description
FACT(1)=>FACT(4)	4F10.0	unit conversion factors for the load cases (1, 2, 3, 4)

APPENDIX 2

SAMPLE DATA FILES

Description of data files

The model of all data files is same type.

Model : Square plate , thickness = 0.1, 10x10 meshes, 4 edges simple support.

Static analysis

DATA FILE # 1 : unit concentrated load at the mid-point of plate

DATA FILE # 2 : uniform pressure loading (p = 0.1)

DATA FILE # 3 : thermal loading without the temperature gradient across the thickness of plate

DATA FILE # 4 : thermal loading with the temperature gradient across the thickness of plate

Eigen value/vector analysis

DATA FILE # 5 : 7 frequencies

DATA FILE # 6 : 10 frequencies, restart (based on the result of the data file # 5)

```
<<COLUMNS>>
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
---------v---------v---------v---------v---------v---------v---------v---------
C DATA FILE # 1 - 4 edge:S.S COND., CONCENTRIC LOADING (10X10 MESH)
       121    1    1    0
    1    1    1    1    0    0    1      0.00      0.00      0.00    0
    2    1    1    1    0    1    1      2.00      0.00      0.00    1
   10    1    1    1    0    1    1     18.00      0.00      0.00    1
   11    1    1    1    0    0    1     20.00      0.00      0.00    0
   12    1    1    1    1    0    1      0.00      2.00      0.00    0
   13    1    1    0    0    0    1      2.00      2.00      0.00    1
   21    1    1    0    0    0    1     18.00      2.00      0.00    1
   22    1    1    1    1    0    1     20.00      2.00      0.00    0
   23    1    1    1    1    0    1      0.00      4.00      0.00    0
   24    1    1    0    0    0    1      2.00      4.00      0.00    1
   32    1    1    0    0    0    1     18.00      4.00      0.00    1
   33    1    1    1    1    0    1     20.00      4.00      0.00    0
   34    1    1    1    1    0    1      0.00      6.00      0.00    0
   35    1    1    0    0    0    1      2.00      6.00      0.00    1
   43    1    1    0    0    0    1     18.00      6.00      0.00    1
   44    1    1    1    1    0    1     20.00      6.00      0.00    0
   45    1    1    1    1    0    1      0.00      8.00      0.00    0
   46    1    1    0    0    0    1      2.00      8.00      0.00    1
   54    1    1    0    0    0    1     18.00      8.00      0.00    1
   55    1    1    1    1    0    1     20.00      8.00      0.00    0
   56    1    1    1    1    0    1      0.00     10.00      0.00    0
   57    1    1    0    0    0    1      2.00     10.00      0.00    1
   65    1    1    0    0    0    1     18.00     10.00      0.00    1
   66    1    1    1    1    0    1     20.00     10.00      0.00    0
   67    1    1    1    1    0    1      0.00     12.00      0.00    0
   68    1    1    0    0    0    1      2.00     12.00      0.00    1
   76    1    1    0    0    0    1     18.00     12.00      0.00    1
   77    1    1    1    1    0    1     20.00     12.00      0.00    0
   78    1    1    1    1    0    1      0.00     14.00      0.00    0
   79    1    1    0    0    0    1      2.00     14.00      0.00    1
   87    1    1    0    0    0    1     18.00     14.00      0.00    1
   88    1    1    1    1    0    1     20.00     14.00      0.00    0
   89    1    1    1    1    0    1      0.00     16.00      0.00    0
   90    1    1    0    0    0    1      2.00     16.00      0.00    1
   98    1    1    0    0    0    1     18.00     16.00      0.00    1
   99    1    1    1    1    0    1     20.00     16.00      0.00    0
  100    1    1    1    1    0    1      0.00     18.00      0.00    0
  101    1    1    0    0    0    1      2.00     18.00      0.00    1
  109    1    1    0    0    0    1     18.00     18.00      0.00    1
  110    1    1    1    1    0    1     20.00     18.00      0.00    0
  111    1    1    1    0    0    1      0.00     20.00      0.00    0
  112    1    1    1    0    1    1      2.00     20.00      0.00    1
  120    1    1    1    0    1    1     18.00     20.00      0.00    1
  121    1    1    1    0    0    1     20.00     20.00      0.00    0
    1    1
   61    3     -1.0
    1  100    1    1
    1
```

```
<<COLUMNS>>
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
---------v---------v---------v---------v---------v---------v---------v---------
1000.     300.       0.0      1000.       0.0      350.

/blank card/
/blank card/
/blank card/
/blank card/
/blank card/
    1    1    2   13   12    0    1    1         0.1
   10   10   11   22   21    0    1    1         0.1
   11   12   13   24   23    0    1    1         0.1
   20   21   22   33   32    0    1    1         0.1
   21   23   24   35   34    0    1    1         0.1
   30   32   33   44   43    0    1    1         0.1
   31   34   35   46   45    0    1    1         0.1
   40   43   44   55   54    0    1    1         0.1
   41   45   46   57   56    0    1    1         0.1
   50   54   55   66   65    0    1    1         0.1
   51   56   57   68   67    0    1    1         0.1
   60   65   66   77   76    0    1    1         0.1
   61   67   68   79   78    0    1    1         0.1
   70   76   77   88   87    0    1    1         0.1
   71   78   79   90   89    0    1    1         0.1
   80   87   88   99   98    0    1    1         0.1
   81   89   90  101  100    0    1    1         0.1
   90   98   99  110  109    0    1    1         0.1
   91  100  101  112  111    0    1    1         0.1
  100  109  110  121  120    0    1    1         0.1
/blank card/
/blank card/
```

```
<<COLUMNS>>
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
---------v---------v---------v---------v---------v---------v---------v---------
C DATA FILE # 2 - 4 edge:S.S COND., PRESSURE LOADING (10X10 MESH)
       121    1    1    0
    1    1    1    1    0    0    1      0.00      0.00      0.00    0
    2    1    1    1    0    1    1      2.00      0.00      0.00    1
   10    1    1    1    0    1    1     18.00      0.00      0.00    1
   11    1    1    1    0    0    1     20.00      0.00      0.00    0
   12    1    1    1    1    0    1      0.00      2.00      0.00    0
   13    1    1    0    0    0    1      2.00      2.00      0.00    1
   21    1    1    0    0    0    1     18.00      2.00      0.00    1
   22    1    1    1    1    0    1     20.00      2.00      0.00    0
   23    1    1    1    1    0    1      0.00      4.00      0.00    0
   24    1    1    0    0    0    1      2.00      4.00      0.00    1
   32    1    1    0    0    0    1     18.00      4.00      0.00    1
   33    1    1    1    1    0    1     20.00      4.00      0.00    0
   34    1    1    1    1    0    1      0.00      6.00      0.00    0
   35    1    1    0    0    0    1      2.00      6.00      0.00    1
   43    1    1    0    0    0    1     18.00      6.00      0.00    1
   44    1    1    1    1    0    1     20.00      6.00      0.00    0
   45    1    1    1    1    0    1      0.00      8.00      0.00    0
   46    1    1    0    0    0    1      2.00      8.00      0.00    1
   54    1    1    0    0    0    1     18.00      8.00      0.00    1
   55    1    1    1    1    0    1     20.00      8.00      0.00    0
   56    1    1    1    1    0    1      0.00     10.00      0.00    0
   57    1    1    0    0    0    1      2.00     10.00      0.00    1
   65    1    1    0    0    0    1     18.00     10.00      0.00    1
   66    1    1    1    1    0    1     20.00     10.00      0.00    0
   67    1    1    1    1    0    1      0.00     12.00      0.00    0
   68    1    1    0    0    0    1      2.00     12.00      0.00    1
   76    1    1    0    0    0    1     18.00     12.00      0.00    1
   77    1    1    1    1    0    1     20.00     12.00      0.00    0
   78    1    1    1    1    0    1      0.00     14.00      0.00    0
   79    1    1    0    0    0    1      2.00     14.00      0.00    1
   87    1    1    0    0    0    1     18.00     14.00      0.00    1
   88    1    1    1    1    0    1     20.00     14.00      0.00    0
   89    1    1    1    1    0    1      0.00     16.00      0.00    0
   90    1    1    0    0    0    1      2.00     16.00      0.00    1
   98    1    1    0    0    0    1     18.00     16.00      0.00    1
   99    1    1    1    1    0    1     20.00     16.00      0.00    0
  100    1    1    1    1    0    1      0.00     18.00      0.00    0
  101    1    1    0    0    0    1      2.00     18.00      0.00    1
  109    1    1    0    0    0    1     18.00     18.00      0.00    1
  110    1    1    1    1    0    1     20.00     18.00      0.00    0
  111    1    1    1    0    0    1      0.00     20.00      0.00    0
  112    1    1    1    0    1    1      2.00     20.00      0.00    1
  120    1    1    1    0    1    1     18.00     20.00      0.00    1
  121    1    1    1    0    0    1     20.00     20.00      0.00    0
    1    1
   61    3       0.0
    1  100    1    1
    1
```

```
<<COLUMNS>>
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
---------v---------v---------v---------v---------v---------v---------v----------
     1000.      300.       0.0     1000.       0.0      350.
 -1.0
/blank card/
/blank card/
/blank card/
/blank card/
    1    1    2   13   12    0    1    1       0.1       0.1
   10   10   11   22   21    0    1    1       0.1       0.1
   11   12   13   24   23    0    1    1       0.1       0.1
   20   21   22   33   32    0    1    1       0.1       0.1
   21   23   24   35   34    0    1    1       0.1       0.1
   30   32   33   44   43    0    1    1       0.1       0.1
   31   34   35   46   45    0    1    1       0.1       0.1
   40   43   44   55   54    0    1    1       0.1       0.1
   41   45   46   57   56    0    1    1       0.1       0.1
   50   54   55   66   65    0    1    1       0.1       0.1
   51   56   57   68   67    0    1    1       0.1       0.1
   60   65   66   77   76    0    1    1       0.1       0.1
   61   67   68   79   78    0    1    1       0.1       0.1
   70   76   77   88   87    0    1    1       0.1       0.1
   71   78   79   90   89    0    1    1       0.1       0.1
   80   87   88   99   98    0    1    1       0.1       0.1
   81   89   90  101  100    0    1    1       0.1       0.1
   90   98   99  110  109    0    1    1       0.1       0.1
   91  100  101  112  111    0    1    1       0.1       0.1
  100  109  110  121  120    0    1    1       0.1       0.1
       1.0
/blank card/
```

```
<<COLUMNS>>
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
---------v---------v---------v---------v---------v---------v---------v---------
C DATA FILE # 3 - 4 edge:S.S COND., THERMAL LOADING (10X10 MESH)
      121    1    1    0
    1    1    1    1    0    0    1     0.00      0.00      0.00   0
    2    1    1    1    0    1    1     2.00      0.00      0.00   1
   10    1    1    1    0    1    1    18.00      0.00      0.00   1
   11    1    1    1    0    0    1    20.00      0.00      0.00   0
   12    1    1    1    1    0    1     0.00      2.00      0.00   0
   13    0    0    0    0    0    1     2.00      2.00      0.00   1
   21    0    0    0    0    0    1    18.00      2.00      0.00   1
   22    1    1    1    1    0    1    20.00      2.00      0.00   0
   23    1    1    1    1    0    1     0.00      4.00      0.00   0
   24    0    0    0    0    0    1     2.00      4.00      0.00   1
   32    0    0    0    0    0    1    18.00      4.00      0.00   1
   33    1    1    1    1    0    1    20.00      4.00      0.00   0
   34    1    1    1    1    0    1     0.00      6.00      0.00   0
   35    0    0    0    0    0    1     2.00      6.00      0.00   1
   43    0    0    0    0    0    1    18.00      6.00      0.00   1
   44    1    1    1    1    0    1    20.00      6.00      0.00   0
   45    1    1    1    1    0    1     0.00      8.00      0.00   0
   46    0    0    0    0    0    1     2.00      8.00      0.00   1
   54    0    0    0    0    0    1    18.00      8.00      0.00   1
   55    1    1    1    1    0    1    20.00      8.00      0.00   0
   56    1    1    1    1    0    1     0.00     10.00      0.00   0
   57    0    0    0    0    0    1     2.00     10.00      0.00   1
   65    0    0    0    0    0    1    18.00     10.00      0.00   1
   66    1    1    1    1    0    1    20.00     10.00      0.00   0
   67    1    1    1    1    0    1     0.00     12.00      0.00   0
   68    0    0    0    0    0    1     2.00     12.00      0.00   1
   76    0    0    0    0    0    1    18.00     12.00      0.00   1
   77    1    1    1    1    0    1    20.00     12.00      0.00   0
   78    1    1    1    1    0    1     0.00     14.00      0.00   0
   79    0    0    0    0    0    1     2.00     14.00      0.00   1
   87    0    0    0    0    0    1    18.00     14.00      0.00   1
   88    1    1    1    1    0    1    20.00     14.00      0.00   0
   89    1    1    1    1    0    1     0.00     16.00      0.00   0
   90    0    0    0    0    0    1     2.00     16.00      0.00   1
   98    0    0    0    0    0    1    18.00     16.00      0.00   1
   99    1    1    1    1    0    1    20.00     16.00      0.00   0
  100    1    1    1    1    0    1     0.00     18.00      0.00   0
  101    0    0    0    0    0    1     2.00     18.00      0.00   1
  109    0    0    0    0    0    1    18.00     18.00      0.00   1
  110    1    1    1    1    0    1    20.00     18.00      0.00   0
  111    1    1    1    0    0    1     0.00     20.00      0.00   0
  112    1    1    1    0    1    1     2.00     20.00      0.00   1
  120    1    1    1    0    1    1    18.00     20.00      0.00   1
  121    1    1    1    0    0    1    20.00     20.00      0.00   0
    1    1
   61    3       0.0
    1  100    1    1
    1
```

```
<<COLUMNS>>
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
---------v---------v---------v---------v---------v---------v---------v---------
     1000.      300.       0.0     1000.       0.0      350.
/blank card/
  1.0
/blank card/
/blank card/
/blank card/
    1    1    2   13   12    0    1    1       0.1       0.0       0.5
   10   10   11   22   21    0    1    1       0.1       0.0       0.5
   11   12   13   24   23    0    1    1       0.1       0.0
   20   21   22   33   32    0    1    1       0.1       0.0
   21   23   24   35   34    0    1    1       0.1       0.0
   30   32   33   44   43    0    1    1       0.1       0.0
   31   34   35   46   45    0    1    1       0.1       0.0
   40   43   44   55   54    0    1    1       0.1       0.0
   41   45   46   57   56    0    1    1       0.1       0.0
   50   54   55   66   65    0    1    1       0.1       0.0
   51   56   57   68   67    0    1    1       0.1       0.0
   60   65   66   77   76    0    1    1       0.1       0.0
   61   67   68   79   78    0    1    1       0.1       0.0
   70   76   77   88   87    0    1    1       0.1       0.0
   71   78   79   90   89    0    1    1       0.1       0.0
   80   87   88   99   98    0    1    1       0.1       0.0
   81   89   90  101  100    0    1    1       0.1       0.0
   90   98   99  110  109    0    1    1       0.1       0.0
   91  100  101  112  111    0    1    1       0.1       0.0
  100  109  110  121  120    0    1    1       0.1       0.0
       1.0
/blank card/
```

```
<<COLUMNS>>
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
---------v---------v---------v---------v---------v---------v---------v---------C
DATA FILE # 4 - 4 edge:S.S COND., THERMAL LOADING (10X10 MESH)
       121    1    1    0
    1    1    1    1    0    0    1      0.00      0.00      0.00    0
    2    1    1    1    0    1    1      2.00      0.00      0.00    1
   10    1    1    1    0    1    1     18.00      0.00      0.00    1
   11    1    1    1    0    0    1     20.00      0.00      0.00    0
   12    1    1    1    1    0    1      0.00      2.00      0.00    0
   13    0    0    0    0    0    1      2.00      2.00      0.00    1
   21    0    0    0    0    0    1     18.00      2.00      0.00    1
   22    1    1    1    1    0    1     20.00      2.00      0.00    0
   23    1    1    1    1    0    1      0.00      4.00      0.00    0
   24    0    0    0    0    0    1      2.00      4.00      0.00    1
   32    0    0    0    0    0    1     18.00      4.00      0.00    1
   33    1    1    1    1    0    1     20.00      4.00      0.00    0
   34    1    1    1    1    0    1      0.00      6.00      0.00    0
   35    0    0    0    0    0    1      2.00      6.00      0.00    1
   43    0    0    0    0    0    1     18.00      6.00      0.00    1
   44    1    1    1    1    0    1     20.00      6.00      0.00    0
   45    1    1    1    1    0    1      0.00      8.00      0.00    0
   46    0    0    0    0    0    1      2.00      8.00      0.00    1
   54    0    0    0    0    0    1     18.00      8.00      0.00    1
   55    1    1    1    1    0    1     20.00      8.00      0.00    0
   56    1    1    1    1    0    1      0.00     10.00      0.00    0
   57    0    0    0    0    0    1      2.00     10.00      0.00    1
   65    0    0    0    0    0    1     18.00     10.00      0.00    1
   66    1    1    1    1    0    1     20.00     10.00      0.00    0
   67    1    1    1    1    0    1      0.00     12.00      0.00    0
   68    0    0    0    0    0    1      2.00     12.00      0.00    1
   76    0    0    0    0    0    1     18.00     12.00      0.00    1
   77    1    1    1    1    0    1     20.00     12.00      0.00    0
   78    1    1    1    1    0    1      0.00     14.00      0.00    0
   79    0    0    0    0    0    1      2.00     14.00      0.00    1
   87    0    0    0    0    0    1     18.00     14.00      0.00    1
   88    1    1    1    1    0    1     20.00     14.00      0.00    0
   89    1    1    1    1    0    1      0.00     16.00      0.00    0
   90    0    0    0    0    0    1      2.00     16.00      0.00    1
   98    0    0    0    0    0    1     18.00     16.00      0.00    1
   99    1    1    1    1    0    1     20.00     16.00      0.00    0
  100    1    1    1    1    0    1      0.00     18.00      0.00    0
  101    0    0    0    0    0    1      2.00     18.00      0.00    1
  109    0    0    0    0    0    1     18.00     18.00      0.00    1
  110    1    1    1    1    0    1     20.00     18.00      0.00    0
  111    1    1    1    0    0    1      0.00     20.00      0.00    0
  112    1    1    1    0    1    1      2.00     20.00      0.00    1
  120    1    1    1    0    1    1     18.00     20.00      0.00    1
  121    1    1    1    0    0    1     20.00     20.00      0.00    0
    1    1
   61    3         0.0
    1  100    1    1
    1
```

```
<<COLUMNS>>
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
---------v---------v---------v---------v---------v---------v---------v---------C
     1000.      300.       0.0     1000.       0.0      350.
/blank card/
  1.0
/blank card/
/blank card/
/blank card/
    1    1    2   13   12    0    1    1       0.1       0.0       0.5       0.2
   10   10   11   22   21    0    1    1       0.1       0.0       0.5       0.2
   11   12   13   24   23    0    1    1       0.1       0.0
   20   21   22   33   32    0    1    1       0.1       0.0
   21   23   24   35   34    0    1    1       0.1       0.0
   30   32   33   44   43    0    1    1       0.1       0.0
   31   34   35   46   45    0    1    1       0.1       0.0
   40   43   44   55   54    0    1    1       0.1       0.0
   41   45   46   57   56    0    1    1       0.1       0.0
   50   54   55   66   65    0    1    1       0.1       0.0
   51   56   57   68   67    0    1    1       0.1       0.0
   60   65   66   77   76    0    1    1       0.1       0.0
   61   67   68   79   78    0    1    1       0.1       0.0
   70   76   77   88   87    0    1    1       0.1       0.0
   71   78   79   90   89    0    1    1       0.1       0.0
   80   87   88   99   98    0    1    1       0.1       0.0
   81   89   90  101  100    0    1    1       0.1       0.0
   90   98   99  110  109    0    1    1       0.1       0.0
   91  100  101  112  111    0    1    1       0.1       0.0
  100  109  110  121  120    0    1    1       0.1       0.0
       1.0
/blank card/
```

```
<<COLUMNS>>
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
---------v---------v---------v---------v---------v---------v---------v---------
C DATA FILE # 5 - 4 edge:S.S COND., DYNAMIC, 7 FREQUENCIES (10X10 MESH)
    1  121    1    7    0
                                                           1
    1    1    1    1    0    0    1      0.00      0.00      0.00    0
    2    1    1    1    0    1    1      2.00      0.00      0.00    1
   10    1    1    1    0    1    1     18.00      0.00      0.00    1
   11    1    1    1    0    0    1     20.00      0.00      0.00    0
   12    1    1    1    1    0    1      0.00      2.00      0.00    0
   13    1    1    0    0    0    1      2.00      2.00      0.00    1
   21    1    1    0    0    0    1     18.00      2.00      0.00    1
   22    1    1    1    1    0    1     20.00      2.00      0.00    0
   23    1    1    1    1    0    1      0.00      4.00      0.00    0
   24    1    1    0    0    0    1      2.00      4.00      0.00    1
   32    1    1    0    0    0    1     18.00      4.00      0.00    1
   33    1    1    1    1    0    1     20.00      4.00      0.00    0
   34    1    1    1    1    0    1      0.00      6.00      0.00    0
   35    1    1    0    0    0    1      2.00      6.00      0.00    1
   43    1    1    0    0    0    1     18.00      6.00      0.00    1
   44    1    1    1    1    0    1     20.00      6.00      0.00    0
   45    1    1    1    1    0    1      0.00      8.00      0.00    0
   46    1    1    0    0    0    1      2.00      8.00      0.00    1
   54    1    1    0    0    0    1     18.00      8.00      0.00    1
   55    1    1    1    1    0    1     20.00      8.00      0.00    0
   56    1    1    1    1    0    1      0.00     10.00      0.00    0
   57    1    1    0    0    0    1      2.00     10.00      0.00    1
   65    1    1    0    0    0    1     18.00     10.00      0.00    1
   66    1    1    1    1    0    1     20.00     10.00      0.00    0
   67    1    1    1    1    0    1      0.00     12.00      0.00    0
   68    1    1    0    0    0    1      2.00     12.00      0.00    1
   76    1    1    0    0    0    1     18.00     12.00      0.00    1
   77    1    1    1    1    0    1     20.00     12.00      0.00    0
   78    1    1    1    1    0    1      0.00     14.00      0.00    0
   79    1    1    0    0    0    1      2.00     14.00      0.00    1
   87    1    1    0    0    0    1     18.00     14.00      0.00    1
   88    1    1    1    1    0    1     20.00     14.00      0.00    0
   89    1    1    1    1    0    1      0.00     16.00      0.00    0
   90    1    1    0    0    0    1      2.00     16.00      0.00    1
   98    1    1    0    0    0    1     18.00     16.00      0.00    1
   99    1    1    1    1    0    1     20.00     16.00      0.00    0
  100    1    1    1    1    0    1      0.00     18.00      0.00    0
  101    1    1    0    0    0    1      2.00     18.00      0.00    1
  109    1    1    0    0    0    1     18.00     18.00      0.00    1
  110    1    1    1    1    0    1     20.00     18.00      0.00    0
  111    1    1    1    0    0    1      0.00     20.00      0.00    0
  112    1    1    1    0    1    1      2.00     20.00      0.00    1
  120    1    1    1    0    1    1     18.00     20.00      0.00    1
  121    1    1    1    0    0    1     20.00     20.00      0.00    0

    1  100    1    1
    1
```

```
<<COLUMNS>>
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
---------v---------v---------v---------v---------v---------v---------v---------
     1000.     300.        0.0      1000.       0.0      350.
/blank card/
/blank card/
/blank card/
/blank card/
/blank card/
    1    1    2   13   12    0    1    1        0.1
   10   10   11   22   21    0    1    1        0.1
   11   12   13   24   23    0    1    1        0.1
   20   21   22   33   32    0    1    1        0.1
   21   23   24   35   34    0    1    1        0.1
   30   32   33   44   43    0    1    1        0.1
   31   34   35   46   45    0    1    1        0.1
   40   43   44   55   54    0    1    1        0.1
   41   45   46   57   56    0    1    1        0.1
   50   54   55   66   65    0    1    1        0.1
   51   56   57   68   67    0    1    1        0.1
   60   65   66   77   76    0    1    1        0.1
   61   67   68   79   78    0    1    1        0.1
   70   76   77   88   87    0    1    1        0.1
   71   78   79   90   89    0    1    1        0.1
   80   87   88   99   98    0    1    1        0.1
   81   89   90  101  100    0    1    1        0.1
   90   98   99  110  109    0    1    1        0.1
   91  100  101  112  111    0    1    1        0.1
  100  109  110  121  120    0    1    1        0.1
/blank card/
/blank card/
```

```
<<COLUMNS>>
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
---------v---------v---------v---------v---------v---------v---------v---------
C DATA FILE # 6 - 4 edge:S.S COND. ,RESTART , 10 FREQUENCIES (10X10 MESH)
    1  121    1   10    0
                                                          -1     1

/blank card/
/blank card/
```

Capabilities of SAPNEW

4. Solution Schemes

   a) Static

      - $[L]\,[D]\,[L]^T$ factorization with forward & backward substitutions
      - Preconditioned conjugate gradient scheme
        - Global
        - Element-by-element

   b) Vibration

      - Subspace iteration with restart (permits diagonal mass matrix)
      - Preconditioned conjugate gradient scheme

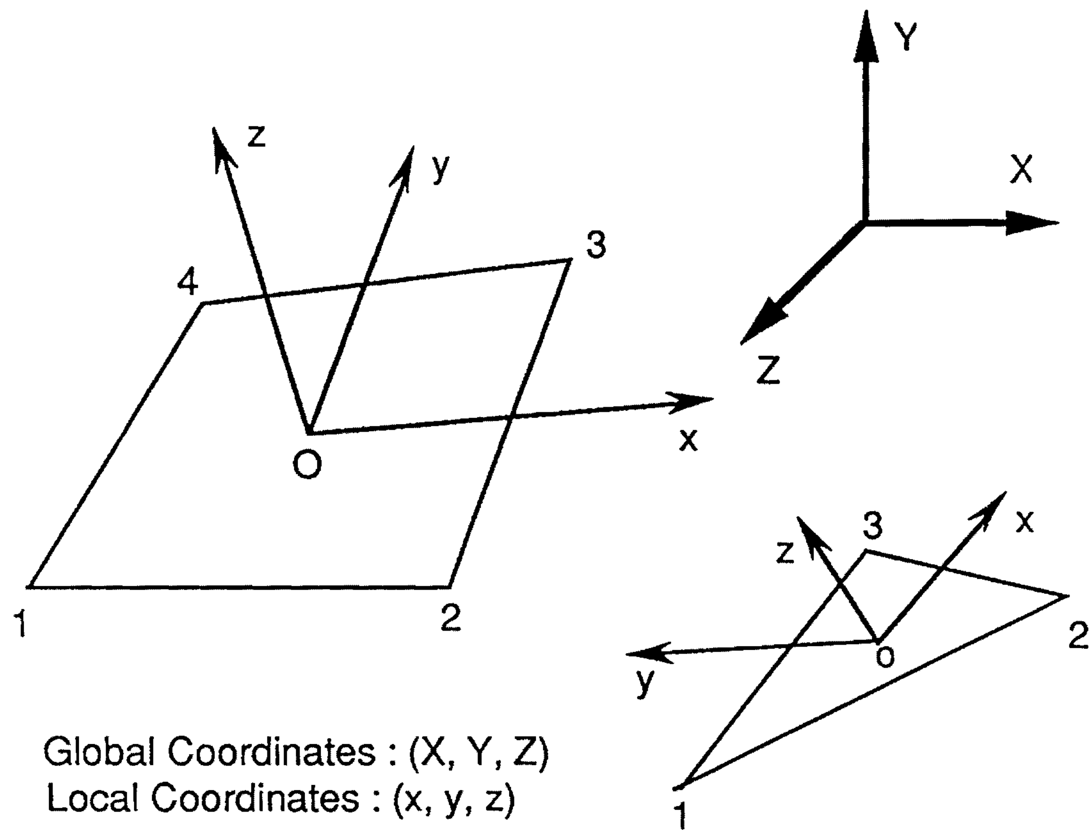


Figure 1. Local coordinate system of the shell element.

Capabilities of SAPNEW

1. Element Library

   - Fully conforming triangular and/or quadrilateral thin shell element (stretching and bending)

2. Loading

   - Concentrated forces (in-plane & transverse) and moments
   - Distributed in-plane edge loading
   - Transverse pressure loading
   - Thermal loading
   - Gravity loading

3. Analysis Capability

   - Static
   - Vibrations (consistent mass matrix with lumped non-structural masses)

   Skyline storage schemes for [K] & [M]

Table 1. Description of models

Model no	1	2	3	4	5	6	7	8
Aspect ratio (b/a)	1.0	1.0	1.0	1.0	1.4	1.4	1.4	1.4
Mesh size	10x10	20x20	30x30	50x50	10x10	20x20	30x30	50x50
Total D.O.F	287	1167	2649	7409	287	1167	2649	7409
Mean bandwidth	30	61	96	156	30	61	96	156

Notes: boundary condition : simple supports on all four sides
plate length : a = 20.0 m
bending rigidity : 0.8333e-1 N-m
mass density : 1.0e-4 kg (mass)
loading type
- Concentrated load applied at mid-point of plate. (F = 1.0 N)
- Uniform pressure load (p = 0.1 N/m2)

Table 2. The results of static analysis

Aspect ratio of shell element	Loading type	Mesh size	Maximum deflection (mm)	theory (mm)	relative error(%)	speed-up ratio
1.0	F	10x10	55.007	55.903	1.60	4.82
		20x20	55.484		0.74	4.94
		30x30	55.623		0.50	5.58
		50x50	55.847		0.10	8.01
	p	10x10	764.31	782.65	2.34	4.91
		20x20	776.04		0.84	4.93
		30x30	779.51		0.41	5.62
		50x50	781.08		0.11	7.86
1.4	F	10x10	70.329	71.518	1.66	4.92
		20x20	71.050		0.65	4.98
		30x30	71.303		0.31	5.60
		50x50	71.374		0.20	8.12
	p	10x10	1333.4	1359.04	1.88	4.87
		20x20	1353.5		0.41	5.01
		30x30	1361.1		0.15	5.80
		50x50	1358.9		0.10	8.09

Notes: F : concentrated load at the mid-point of plate
p : uniform pressure load
speed-up ratio : execution time of sequential code is divided by the execution time of vectorization & parallelization code

Table 3. The results of the dynamic analysis

Model no.		Frequencies of modes (Hz)							speed-up ratio
		1	2	3	4	5	6	7	
1	C	4.5717	11.331	11.331	18.216	22.776	22.776	29.777	8.0
	T	4.5048	11.262	11.262	18.019	22.524	22.524	29.281	
	E	1.5	0.6	0.6	1.1	1.1	1.1	1.7	
2	C	4.5079	11.279	11.279	18.069	22.587	22.587	29.406	8.0
	T	4.5048	11.262	11.262	18.019	22.524	22.524	29.281	
	E	0.06	0.15	0.15	0.28	0.27	0.27	0.4	
3	C	4.5061	11.269	11.269	18.041	22.551	22.551	29.336	10.42
	T	4.5048	11.262	11.262	18.019	22.524	22.524	29.281	
	E	0.02	0.06	0.06	0.12	0.1	0.1	0.18	
4	C	4.5053	11.264	11.264	18.027	22.534	22.534	29.301	12.89
	T	4.5048	11.262	11.262	18.019	22.524	22.524	29.281	
	E	0.01	0.02	0.02	0.04	0.04	0.04	0.68	
5	C	3.4594	6.9313	10.291	13.208	19.564	20.845	27.752	8.0
	T	3.4016	6.8492	10.159	13.065	19.352	20.639	27.396	
	E	1.7	1.2	1.3	1.1	1.1	1.0	1.3	
6	C	3.4458	6.9176	10.230	13.143	19.352	20.701	27.451	8.0
	T	3.4016	6.8492	10.159	13.065	19.352	20.639	27.396	
	E	1.3	1.0	0.7	0.6	0.8	0.3	0.2	
7	C	3.4390	6.9245	10.230	13.104	19.448	20.680	27.451	10.43
	T	3.4016	6.8492	10.159	13.065	19.352	20.639	27.396	
	E	1.1	1.1	0.7	0.3	0.5	0.2	0.2	
8	C	3.4322	6.8971	10.169	13.130	19.390	20.680	27.478	14.87
	T	3.4016	6.8492	10.159	13.065	19.352	20.639	27.396	
	E	0.9	0.7	0.1	0.5	0.2	0.2	0.3	

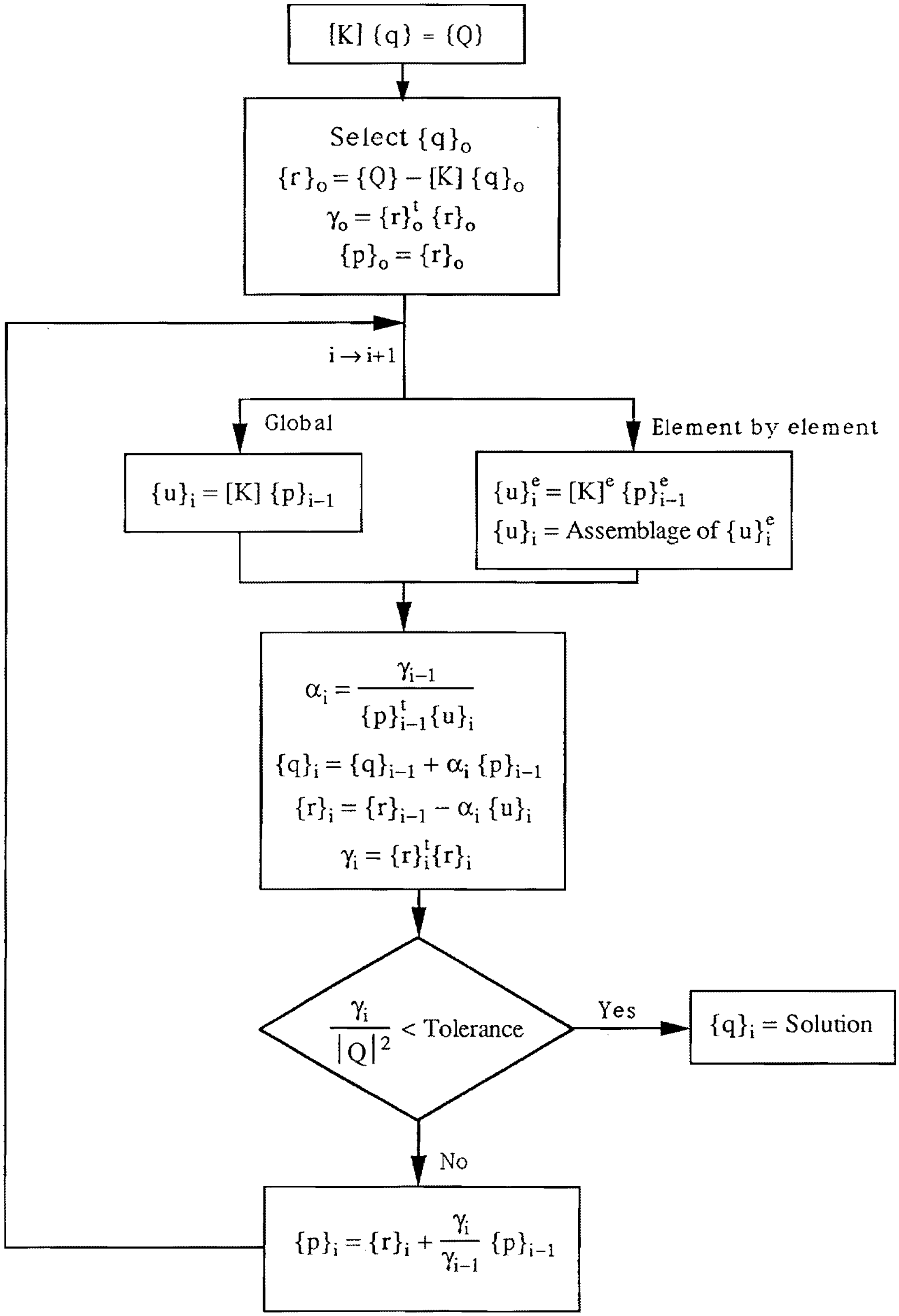
Notes: model no. : refer to Table 1.
C : calculated value
T : theoretical value[6]
E : relative error (%)
speed-up ratio : see the definition of Table 2.

Table 4. The comparison of the execution times for the restart option in the eigen value/vector analysis

Mesh size	Execution time (sec)			
	sequential.	vect.†	vect. & parallel (initial execution)	vect. & parallel (restart)
20x20	247.48	60.58	30.97	12.35
30x30	861.25	170.32	82.60	31.26
50x50	4952.12	690.15	322.62	173.31

Notes: Total modes : 7
modal information of restart : previously evaluated 5 modes
† Vectorization with global optimization.

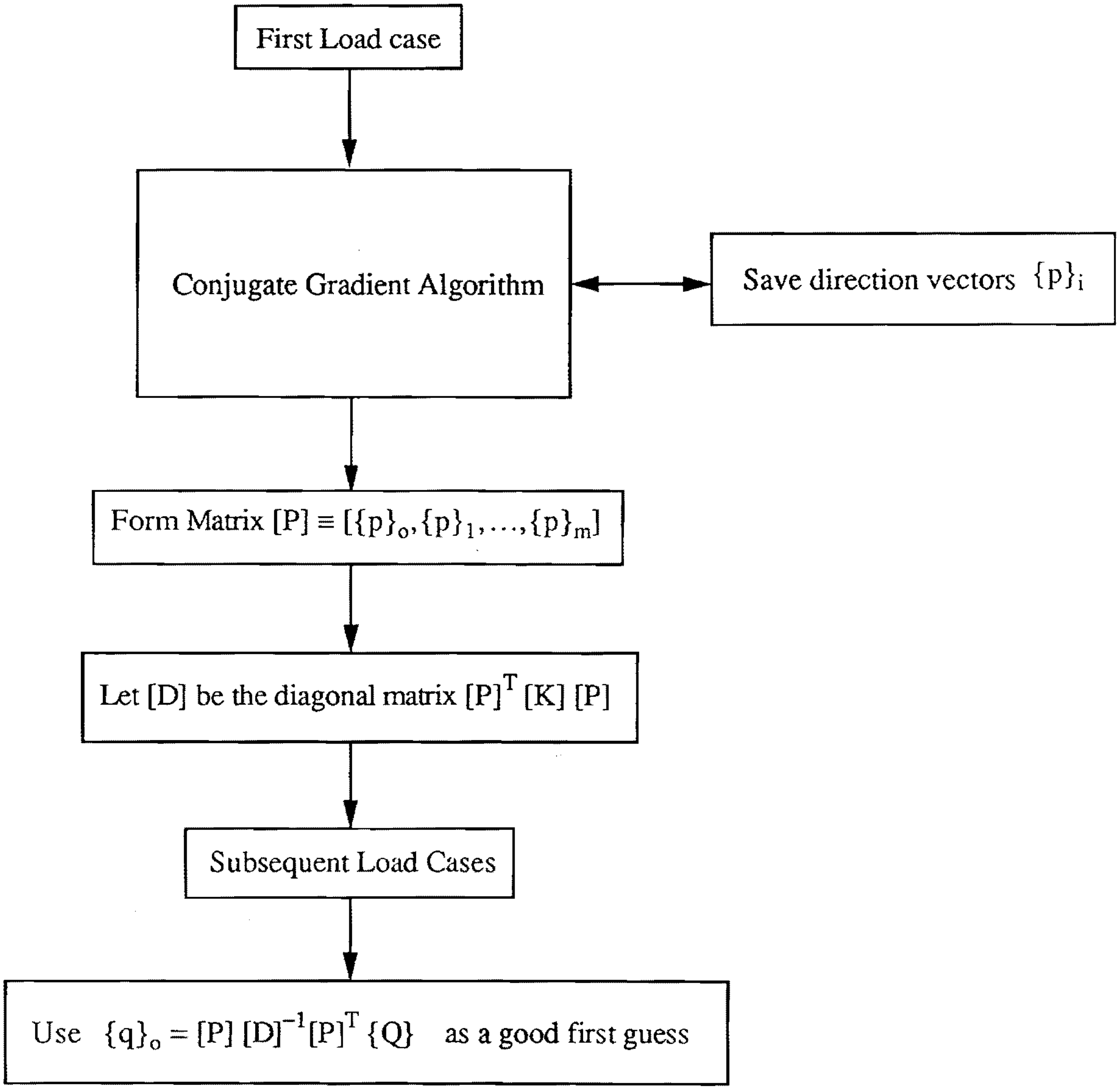
Conjugate Gradient Algorithm for Static Analysis



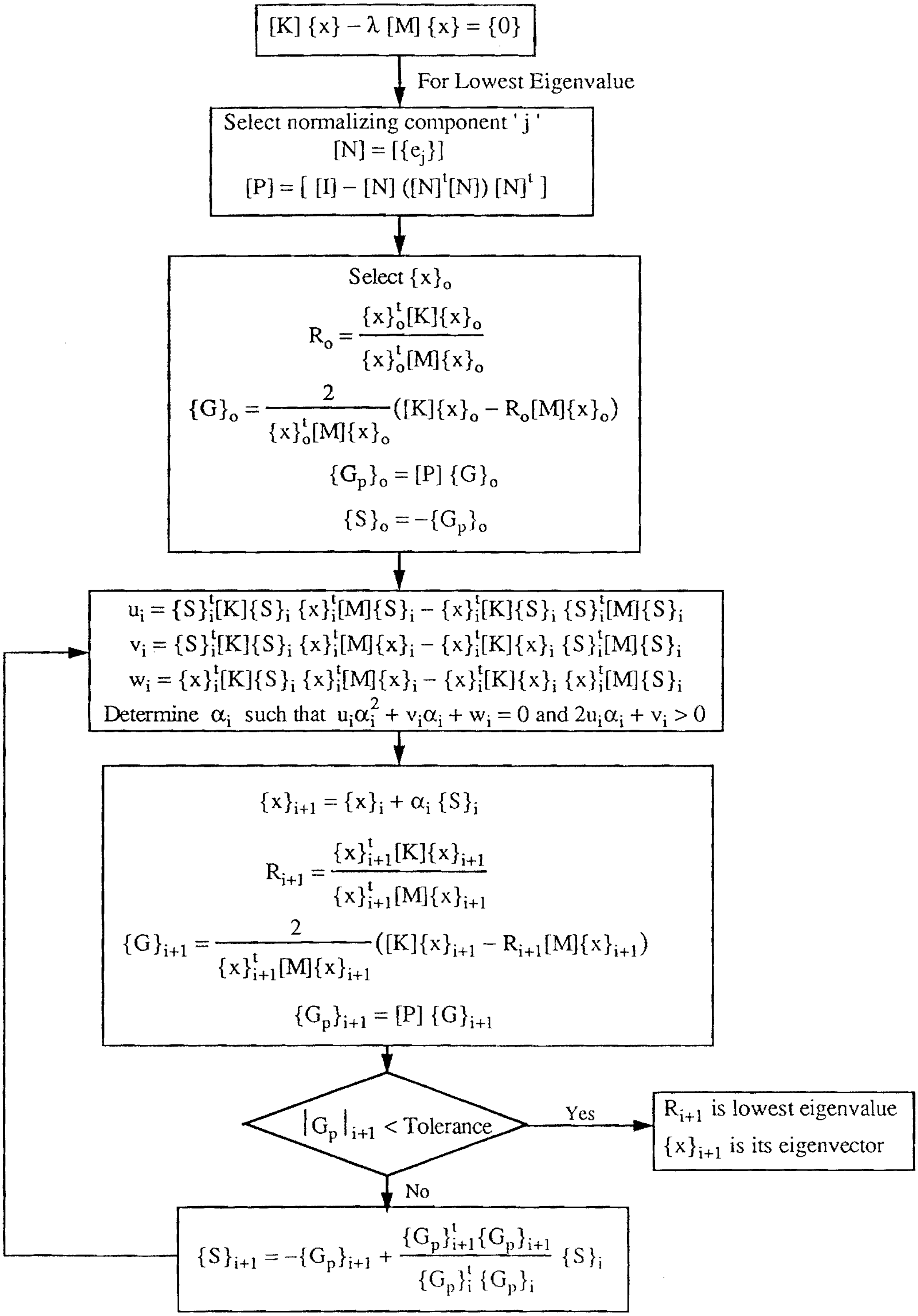
Portion of Code to Calculate the Product [K]{p}

```
      DO 120 ETYP=1,NTYP
        EDOF=NDOF*ENOD(ETYP)
        IBEGIN=IFEL(ETYP)
        IEND=IFEL(ETYP)+NEL(ETYP)-1
      DO 106 I=IBEGIN,IEND
        J=1
        K2=LM(J,I)
        U1(J,I)=0.
        IF(K2.NE.0) THEN
        NPNT=NPNTK(I)
        IJJJ=IJ(J)
cvd$  shortloop
        DO 105 K=1,EDOF
          K1=LM(K,I)
          U1(J,I)=U1(J,I)+UN(NPNT+IJJJ+K)*P(K1)
105     CONTINUE
        ENDIF
106   CONTINUE
      DO 113 J=2,EDOF
        DO 112 I=IBEGIN,IEND
          K2=LM(J,I)
PARALLELIZED  U1(J,I)=0.
          IF(K2.NE.0) THEN
          NPNT=NPNTK(I)
cvd$  shortloop
          DO 7110 KI=1,J-1
            KII=LM(KI,I)
VECTORIZED  U1(J,I)=U1(J,I)+UN(NPNT+IJ(KI)+J)*P(KII)
7110      CONTINUE
          ENDIF
112     CONTINUE
ZED
113   CONTINUE
      DO 7120 J=2,EDOF
        IJJJ=IJ(J)
        DO 115 I=IBEGIN,IEND
          K2=LM(J,I)
PARALLELIZED  U2(J,I)=0.
          IF(K2.NE.0) THEN
          NPNT=NPNTK(I)
cvd$  shortloop
          DO 114 KJ=J,EDOF
            KJJ=LM(KJ,I)
VECTORIZED  U2(J,I)=U2(J,I)+UN(NPNT+IJJJ+KJ)*P(KJJ)
114       CONTINUE
          ENDIF
115     CONTINUE
7120  CONTINUE
      DO 118 I=IBEGIN,IEND
cvd$  nodepchk
cvd$  shortloop
        DO 118 J=1,EDOF
          JJJ=LM(J,I)
118       U(JJJ)=U(JJJ)+U1(J,I)+U2(J,I)
120   CONTINUE
```

Multiple Load Cases



Conjugate Gradient Algorithm for Dynamic Analysis



Future Work

1. Implement schemes to accept NASTRAN input.

2. Structure the output to be compatible with the Aeroelastic module.

3. Couple the code with the Aeroelastic module to perform flutter analysis of bladed-disk assemblies.

4. Fine tune EBE PCG method for static and vibration analysis.

5. Implement the Parallel QR algorithm for eigenvalue analysis.

6. Extend element library and/or analysis capabilities, e.g. include a frame element or additional material models.

INTERIM PROGRESS REPORT

NASA Grant #NAG-3-895

Period: January 1, 1990 - December 31, 1990

Finite Element Code for Thin Shell Structures - A Preprocessor for Turbine Blade Analysis on the Alliant FX/80

Manohar P. Kamat
Brian Watson

School of Aerospace Engineering
Georgia Institute of Technology
Atlanta, Geogia 30332

1. INTRODUCTION

The objective of the research activity described in this report is to eventually provide a finite element capability for analyzing Turbo Machinery Bladed Disk Assemblies in a vector/parallel processing environment.

Analysis of aircraft turbo fan engines is computationally intensive. Problems involving aeroelastic stability and response of bladed-disk assemblies in aircraft turbo fan engines are among the most difficult problems encountered. Complications in these studies arise from the small differences between individual blades known as mistuning. Previous researchers have come to believe that the static, flutter, and forced response of mistuned turbo machinery blades can be studied by analyzing each blade separately in either a pure bending or a pure torsional motion. Concurrent (parallel) processing seems to offer the greatest promise for such an analysis.

The performance limit of modern day computers with a single processing unit has been estimated at 3 billions of floating point operations per second (3 gigaflops). In view of this limit of a sequential unit, performance rates higher than 3 gigaflops can be achieved only through vectorization and/or parallelization as on Alliant FX/8. Accordingly, the efforts of this critically needed research have been geared towards developing and evaluating parallel finite element methods for static and vibration analysis of multi-degree-of-freedom blade models built-up from flat thin shell, beam, and spring elements in a special purpose code by the acronym SAPNEW.

- Data file for ASTROP program

  The SAPNEW program now generates a data file for the ASTROP program based on the results of the eigen value/vector analysis.

The added capabilities have led to minor changes in the format for the SAPNEW input data file. The appendix describes these changes.

iii) Performance Evaluation of the SAPNEW Program

Test model

The model used for evaluating the SAPNEW program is a model of the SR5 blade. The conversion program was used to convert a NASTRAN model of the blade to the SAPNEW data input format. Figure 1. shows the geometry of the SR5 blade. Table 1. lists the statistics for the blade model. The test case consists of determining the three lowest eigen values and their corresponding mode shapes for the SR5 blade model using geometric stiffness generated by the static solution of the blade loaded by centrifugal forces.

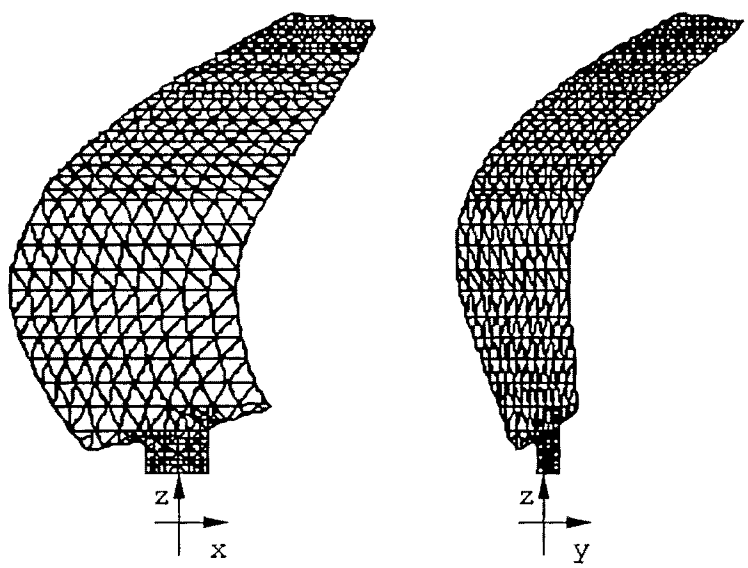


Figure 1. SR5 blade geometry.

discussions with Dr. Murthy of NASA Lewis, it was decided not to improve the initial stress matrix calculation in SAPNEW for the time being.

Table 2. Comparison of results

A. Without effects of centrifugal forces

	MSC/NASTRAN*	SAPNEW	% error
Eigenvalues			
Mode 1	4.334788E+5	4.676620E+5	7.88
Mode 2	2.949795E+6	3.206697E+6	8.71
Mode 3	1.046320E+7	1.082780E+7	3.48
Frequency (Hz)			
Mode 1	104.79	108.84	3.87
Mode 2	273.35	285.00	4.26
Mode 3	514.82	523.71	1.73

Difference in Eigenvectors from MSC/NASTRAN* and SAPNEW : $\cos^{-1}(\phi^T_{Nastran}\ \phi_{Sapnew})$

Mode 1	4.81°
Mode 2	17.28°

B. With the effects of centrifugal forces

	MSC/NASTRAN	SAPNEW	% error
Eigenvalues			
Mode 1	9.0397E+5	12.036 E+5	33.14
Mode 2	3.1197E+6	3.2612E+6	4.54
Mode 3	1.3572E+7	1.2520E+7	- 7.75
Frequency (Hz)			
Mode 1	151.32	174.60	15.38
Mode 2	281.11	287.41	2.24
Mode 3	586.33	563.16	- 3.95

Difference in Eigenvectors from MSC/NASTRAN and SAPNEW : $\cos^{-1}(\phi^T_{Nastran}\ \phi_{Sapnew})$

Mode 1	22.38°
Mode 2	49.82°

Note : * Results for this model were obtained at Ga. Tech

Results

The time required by the SAPNEW program to complete the test case for different optimization options and different numbers of processors is listed in Table 4. The corresponding speed up values are listed in Table 5. and graphically displayed in Figure 2.

Table 4. Time Results in Seconds

Program Phase	Number of Processors								
	One †	One ††	Two	Three	Four	Five	Six	Seven	Eight
Input	82.09	22.29	19.66	18.58	18.28	18	18.01	17.9	17.96
Static Solution	217.33	30.14	24.85	22.41	21.52	20.76	20.58	20.32	20.21
Eigen Analysis	522.13	63.24	44.53	36.14	33.15	30.39	29.7	28.8	28.47
Printing	1.83	1.62	1.04	0.86	0.83	0.8	0.79	0.78	0.78
Total	823.38	117.29	90.08	77.99	73.78	69.95	69.08	67.8	67.42

Notes: † No Optimization
†† Vectorization

Table 5. Speedup Results

	Number of Processors							
	One	Two	Three	Four	Five	Six	Seven	Eight
Speedup Due to Vectorization and Parallelization	7.02 †	9.14	10.56	11.16	11.77	11.92	12.14	12.21
Speedup Due to Parallelization Only		1.30	1.50	1.59	1.68	1.70	1.73	1.74

Note: † Vectorization Only

3. ONGOING TASKS

i) Substructuring

Background and Motivation

The finite element method applied to the displacement formulation of an elasticity problem yields equations of equilibrium for each element relating the element stiffness matrix, the element nodal displacements, and the element nodal forces.

$$[K^e]\{q^e\} = \{F^e\} \qquad e=1,2,\ldots,\text{Number of elements} \qquad (1)$$

The systems of equations for each element are independent and could, in theory, be solved concurrently. However, in general, this is not possible because the element nodal forces are not known until the nodal displacements are determined.

Accordingly, the requirement that the displacements be continuous across inter-element boundaries is introduced. This allows the element level equilibrium equations (Eq. 1) to be assembled into a global set of equilibrium equations involving the global stiffness matrix, the nodal displacements, and the externally applied nodal forces.

$$[K]\{q\} = \{F\} \qquad (2)$$

However, in so doing, the inherent parallelism of the elements is lost.

To take advantage of the parallel processing architectures, alternate solution methodologies must be used.

Substructuring

One possible way to take advantage of parallel processing is to partially assemble small groups of elements into substuctures. These substructures would

ii) Non-linear solution algorithm

It has been shown that the element-by-element conjugate gradient algorithm offers almost no advantages over the LDL^T algorithm for the linear problems investigated on the Alliant FX/8. However, there may be some merit in the algorithm in solving non-linear problems with load incrementation.

To investigate this, a program for analyzing three-dimensional truss structures with large deflections with a quasi-Newton iteration scheme has been implemented. The quasi-Newton scheme involves the successive solution of similar linear systems. The conjugate gradient algorithm would appear to be very effective in this situation. It is possible to store information about a system (in the form of conjugate direction vectors) and use this information to generate a good initial guess for the next system.

Initial studies have shown that this procedure is very effective in reducing the number of iterations for the conjugate gradient algorithm to converge for each Newton step (See Figure 3.). However, using the LDL^T algorithm to solve each step has remained more cost-effective.

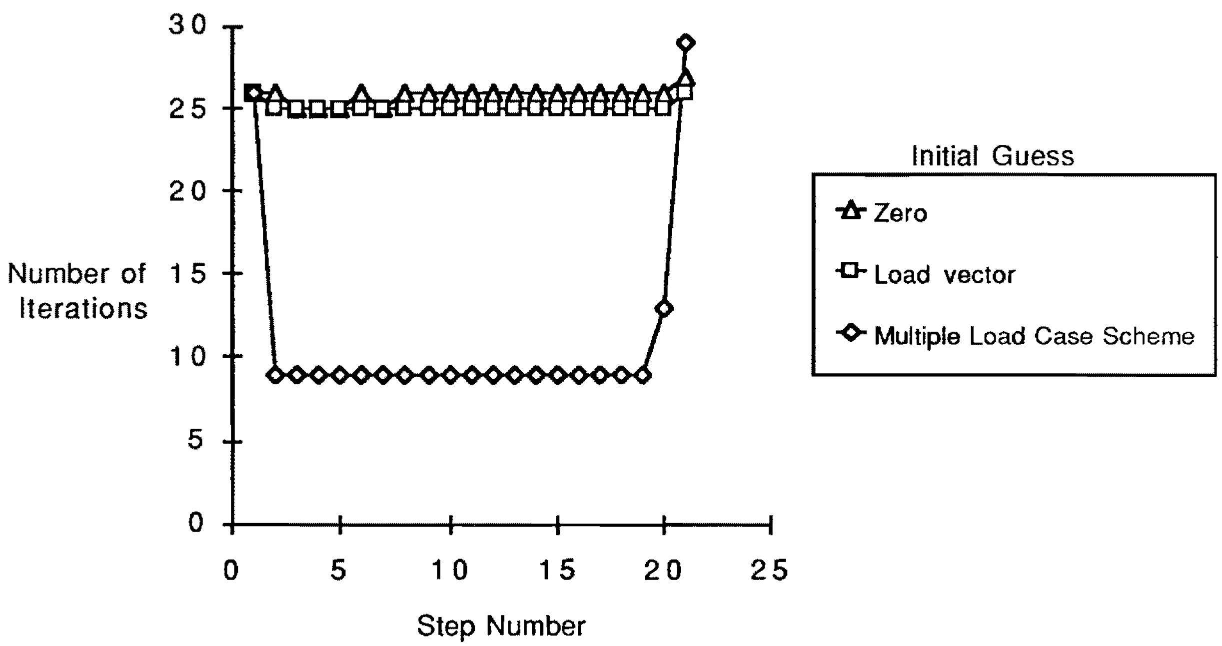


Figure 3. Number of Iterations Required by the Element-by-Element Preconditioned Conjugate Gradient Algorithm to Solve Each Step in the Quasi-Newton Scheme.

5. APPENDIX

Changes to SAPNEW Input Data File Format

Header Card (20A4/8I5)

Variable	Format	Desription
HED	20A4	Title of analysis
NDYN	I5	Analysis code =0; Static Analysis >0; Eigen Analysis NDYN-1 = number of static solution iterations for geometric stiffness computation
NUMNP	I5	Number of node points
NUMEG	I5	Number of element groups
NLCASE	I5	Number of load cases or modes NDYN=0; Load cases NDYN>0; Modes
NSPRNG	I5	Number of spring elements
MODEX	I5	Flag for execution mode =0; Execute =1; Input data verification
IRFCE	I5	Flag for centrifugal force load =0; Do not use =1; Use
MNASTY	I5	Flag for ASTROP data file output =0; Do not make data file =1; Make data file

Beam element control card (5I5)

Variable	Format	Desription
NPAR(1)	I5	Index for element type
		=2; beam element
NPAR(2)	I5	Number of beam elements
NPAR(3)	I5	Number of geometric property sets
NPAR(4)	I5	Number of fixed-end force sets
NPAR(5)	I5	Number of material property sets

Beam element material property set cards (I5,4F10.0)

Variable	Format	Desription
K	I5	Beam material property set number
E	F10.0	Young's modulus
G	F10.0	Poisson's ratio
RO	F10.0	Mass density
WGHT	F10.0	Weight per unit length

Beam element data cards (10I5,2I6,I8)

Variable	Format	Desription
INEL	I5	Element number
INI	I5	Node 1
INJ	I5	Node 2
INK	I5	Node 3
IMAT	I5	Material property set number
IMEL	I5	Geometric property set number
ILC(4)	4I5	End loads
INELKI	I6	End code for node 1
INELKJ	I6	End code for node 2
INC	I8	Element generation code

Note: The beam axis connects nodes 1 & 2. Node 3 detemines the cross section axis 1

Centrifugal force data card (5F10.0)

Variable	Format	Desription
V1	F10.0	X-component of spin axis
V2	F10.0	Y-component of spin axis
V3	F10.0	Z-component of spin axis
OMEGA	F10.0	Spin rate in radians/second
UNITS	F10.0	Unit conversion factor

Note: Spin axis passes through origin

School of Aerospace Engineering

Georgia Institute of Technology
Atlanta, Georgia 30332-0150
404·894·3000
Fax: 404·894·2760

June 19, 1990

Dave Janetzke
NASA Lewis Research Center, MS 23-3
21000 Brookpark Rd
Cleveland, OH 44135

Re: NASA Grant # NAG 3-895 - Semi Annual Progress Report

Dear Dave:

As required by grant regulations, enclosed herewith, please find three copies of a very brief semi-annual progress report on the above grant. If you have any questions, please do not hesitate to call me at (404) 894-7439. Thank you.

Best Regards,

Manohar P. Kamat
Principal Investigator

MPK/jr

SEMI-ANNUAL PROGRESS REPORT

NASA LEWIS GRANT #NAG3-895

Title: **Parallel Processing for the Analysis of Bladed Disk Assemblies**

Period: **January 1, 1990 - June 30, 1990**

Technical Progress

I. The following modifications and/or additional capabilities were implemented into the SAPNEW Program on the Alliant FX/80.

1. Centrifugal force calculation on the basis of the currently used lumped mass formulation in SAPNEW has been implemented;

2. A general 3-D frame element and a spring element has been added to the element library;

3. A multi-point constraint capability has been added and finally

4. A preprocessor has been developed that converts a NASTRAN input deck into an equivalent input deck for SAPNEW.

II. The modified SAPNEW program has been validated on a sample model obtained from NASA Lewis. SAPNEW currently lacks a laminated composite construction model for the blade in question. Hence initial studies will be restricted to an isotropic blade. We are currently in the process of obtaining the details of the NASTRAN model for such a blade.

III. Studies on improved solution algorithms for equilibrium and eigenvalue analyses are continuing.

Semi-Annual Progress Report

NASA Lewis Grant # NAG3-985

Title: Parallel Processing for the Analysis of Bladed Disk Assemblies

Period: January 1, 1991 - June 30, 1991

Technical Progress

Most of the efforts for this reporting period were devoted to two main activities:

1. Assessing ways to improve the performance of SAPNEW especially with regard to its equations-solution phase through improved parallel processing algorithms. Several different algorithms are being evaluated namely: Gauss-Seidel, Jacobi and the Hughes' element-by-element preconditioned conjugate gradient method. All the results of this investigation are being summarized in a paper to be submitted for Journal publication.

2 Developing a structural optimization capability in a parallel processing environment. Two methods such as the sequential unconstrained minimization technique using penalty functions and a dual method such as the optimality criteria method are being evaluated. The results of this investigation will be presented at the session on "High Performance Computing in Mechanics" at the First U.S. National Congress on Computational Mechanics in Chicago, July 21-24, 1991. These results will also be included as part of a chapter of a AIAA's special publication entitled "Structural Optimization-Status and Promise".

Some work has also been initiated on the development of a concurrent optimization algorithm for obtaining globally optimal solutions for nonconvex functions which are encountered in the tailoring of laminated composite structures.

E16-103

Progress Report

NASA Grant #NAG-3-985

High Performance Computing for the Analysis and Design

of Turbo Machinery Bladed Disk Assemblies

Period: Jan. 1, 1991 - Decemaber 31, 1991

Research Activity and Progress During the 1991 Grant Year

a. Techniques for Global Optimization

Design optimization problems (and analysis problems which can be posed as minimization problems) are among the most computationally difficult. The situation is further complicated when the objective function is not convex. Because most optimization techniques are descent type methods, they fail on non-convex problems because of the presence of local minimum points. It was hoped that through this research effort, improved techniques for global optimization could be found.

Young Transformation

The Young transformation of a function is given by

$$f^*(x^*) = \min_x [f(x) - xx^*]$$

When this transformation is applied twice consecutively to a convex function, this result is the original function. When it is applied twice to a non-convex function, the result is the greatest convex function which is always less than or equal to the original function. It was hoped that this property could be used to convert a non-convex optimization problem to a convex one and greatly simplify the solution procedure. However, further study of the young transformation revealed that its practical application presents a more complicated optimization problem than the original problem.

An efficient technique for global optimization was needed to address the problem of determining that stacking sequence for a given number of plies such that its extension-twist coupling can be maximized subject to no curvatures. Such a laminated composite construction can have beneficial implications on the performance of helicopter rotor blades and perhaps also on the blades of other types of rotors of turbo machines. The problem has multiple optima and an accurate solution of the problem is highly intriguing and requires a technique for unconstrained global minimization of an appropriately constructed auxiliary objective function. Such a technique known as the "filled function method" has been developed by R.GE [1]. An attempt was made to implement GE's technique for the solution of the above problem. The work is continuing and may be continued under funding by a different sponsor since this research activity does not appear to be of current interest to the Structural Dynamics Branch at NASA Lewis.

b. Neural Networks

Neural network provide an alternative to traditional computing. Several authors have suggested using neural networks in the solution of analysis and optimization problems [2]-[4]. Hajela and Berke [3] and Swift and Batill [4] have used neural networks in optimization problems. The methodology that these researchers have suggested involves using a neural network to replace evaluation of the objective function. The network becomes an approximation for the objective function by providing a map of input-to-output. The network is trained with a back-propogation technique so as to closely model the objective as possible. It was hoped that further research into neural networks would provide a way to better integrate them into the optimization process to enable efficient global optimization.

Initial efforts in neural networks began with the development of a neural network simulator on a minicomputer. A program which simulates back propogation (see references 5, 6) was written and initial investigations into network training were performed.

c. Optimal Design Using Concurrent Processing

The computer code OPTSTAT [7] which optimizes truss and semi-monocoque structures using the optimality criteria method was selected for conversion to concurrent processing environment on the BBN Butterfly machine at Georgia Tech. Because of lack of modularity of the OPTSTAT code a great deal of difficulty was experienced in the parallelization of the entire code. As a result, parallelization was restricted to truss element. Only the computationally intensive modules of OPTSTAT such as the equations solver subroutine, strain energy density calculation subroutine were parallelized.

The parallelized code was then evaluated on the well-known 200 bar truss model under three different loading conditions. The speed ups were less than encouraging indicating that the parallelization was less than satisfactory. Further work is necessary in this area.

d. Publications

A paper entitled "Structural Optimization Using Concurrent Processing" by G. Kulkarni, C.Y. Song and M. P. Kamat was presented at the 1st U.S. National Congress on Computational Mechanics held in Chicago, Illinois, July 21-24, 1991. The abstract of the paper was published in the conference proceedings. A copy of the abstract was sent to NASA Lewis.

Another paper which summarized the development of the algorithms for more efficient computations was completed. This paper entitled "A Study of Equation Solvers for Linear and Nonlinear Finite Element Analysis for Parallel Processing Computers" by Brian Watson and M.P. Kamat has been sent for consideration of presentation at the 33rd SDM Conference in Dallas, Texas. The paper will be sent for consideration of publication in an appropriate journal. A copy of the paper was forwarded to NASA Lewis.

Finally, a rough draft of contractor's report describing briefly the SAPNEW code including a users' guide was completed. A copy of the report has been sent to NASA Lewis for their comments.

References

[1]. Ge, R., "A Filled Function Method for Finding a Global Minimizer of a Function of Several Variables", Mathematical Programming, Vol. 46, pp. 191-204, 1990.

[2]. Rehak, D.R., Thewalt, C.R. and Doo, L.B., Neural Network Approaches in Structural Mechanics Computations", Computer Utilization in Structural Engineering. ASCE Structural Congress, p. 168, 1989.

[3]. Hajela, P. and Berke, L., "Neurobiological Models in Structural Analysis and Design", AIAA/ASME/ASCE/AHS/ASC 31st Structures, Structural Dynamics, and Materials Conference, p. 345, 1990.

[4]. Swift, R. and Batill, S., "Applications of Neural Networks to preliminary structural Design", AIAA/ASME/ASCE/AHS/ASC 32nd Structures, Structural Dynamics and Materials Conference, p. 335, 1991.

[5]. Wasserman, P., Neural Computing: Theory and Practice, Van Nostrand Reinhold, New York, 1989.

[6]. McClelland, J. and Rumelhart, D., Explorations in Parallel Distributed Processing, MIT Press, 1988.

[7]. Venkayya, V.B. and Tischler, V.A., "OPTSTAT - A Computer Program for the Optimal Design of Structures subjected to Static Loads", AFWAL Tech. memorandum FBR-79-67, June 1979.

r 16-103

<u>Contractor Report</u>
NASA Grant #NAG-3-895

SAPNEW:
Parallel Finite Element Code for Thin Shell Structures on the Alliant FX/80

Manohar P. Kamat
Brian C. Watson

School of Aerospace Engineering
Georgia Institute of Technology
Atlanta, Georgia 30332

February 1992

Contents

1. INTRODUCTION

This report summarizes the results of a research activity aimed at providing a finite element capability for analyzing turbo-machinery bladed-disk assemblies in a vector/parallel processing environment.

Analysis of aircraft turbo fan engines is very computationally intensive. Problems involving aeroelastic stability and response of bladed-disk assemblies in aircraft turbo fan engines are among the most difficult problems encountered. Complications in these studies arise from the small differences between individual blades known as mistuning. Previous researchers have come to believe that the static, flutter, and forced response of mistuned turbo-machinery blades can be studied by analyzing each blade separately in either a pure bending or a pure torsional motion.[1] However, with the development of thin blades with high sweep, it is necessary to model the coupled behavior. This requires a finite element analysis using shell elements, which is time consuming on a sequential computer. Concurrent (parallel) processing seems to offer the greatest promise for such an analysis.

The performance limit of modern day computers with a single processing unit has been estimated at 3 billions of floating point operations per second (3 gigaflops). In view of this limit of a sequential unit, performance rates higher than 3 gigaflops can be achieved only through vectorization and/or parallelization as on Alliant FX/80. Accordingly, the efforts of this critically needed research have been geared towards developing and evaluating parallel finite element methods for static and vibration analysis. A special purpose code, named with the acronym SAPNEW, performs static and eigen analysis of multi-degree-of-freedom blade models built-up from flat thin shell elements.

SAPNEW grew out of the well-known SAP IV and SAP V codes[2,3]. The flat thin shell element, as well as the beam element in SAPNEW were taken directly from the SAP IV and SAP V codes. These were integrated in a finite element code that uses a skyline storage scheme for the assembled mass and stiffness matrices[4] as well as efficient solution schemes for static and eigen analysis designed to accomodate this compact storage method.

The objective behind this concurrent code development on the Alliant FX/80 was to provide a stand alone capability for static and eigen analysis. The output of this program was designed to easily integrate into the input of another concurrent code, known by the acronym ASTROP, for aeroelastic studies[5]. A preprocessor, which accepts NASTRAN input decks and converts them to SAPNEW format, was added to make SAPNEW more user friendly and more readily used by researchers at NASA Lewis Research Center.

2. DESCRIPTION OF SAPNEW

SAPNEW is a finite element code for static and eigen analysis of three-dimensional, thin shell structures, particularly turbo-machinery blades. Structures may be modeled with triangular or quadrilateral flat elements with uncoupled in-plane and bending stiffnesses. Coupling between the in-plane and bending stiffnesses is achieved through assembling non-coplanar elements. Loading of the structure may be due to concentrated loads, normal pressure, thermal effects, uniform acceleration, and/or centrifugal acceleration.

Static Analysis

Linear static analysis may be performed on a model to generate deformation and stress information.

Eigen Analysis

Eigen value/vector analysis may be performed on a model to generate natural frequencies and mode shapes. This analysis may include geometric stiffening of the model due to applied loads and centrifugal effects.

Shell Element

Stiffness matrices

The primary modeling element of the SAPNEW program is a thin shell element. For details of the formulation of this element, consult reference [6]. A CST (constant strain triangular) element models the in-plane behavior. A CST element has six degrees of freedom. A quadrilateral element is formed by the assembly of four CST elements followed by a static condensation procedure to eliminate the interior node to leave eight degrees of freedom.

The bending behavior is modeled by a partially constrained assemblage of three LCCT (linear curvature compatible triangular) elements. Each LCCT element has ten degrees of freedom. Static condensation eliminates the internal node of the assemblage and the constraint of linearly varying curvature eliminates the mid-side degrees of freedom. The resulting triangular element (designated LCCT-9) has nine degrees of freedom. Normal twisting degrees of freedom are then added for the transformation to global coordinates, although no stiffnesses are associated with these degrees of freedom in the local coordinate system. The quadrilateral element is formed from an assembly of four LCCT-9 elements followed by static condensation to eliminate the internal node.

With the in-plane and bending properties combined, the resulting element has six degrees of freedom at each node (three displacements and three rotations).

In calculating the stiffness matrices, the program may (at user's option) use different constitutive (stress-strain) relationships for the in-plane and the bending behaviors. In this way, material properties typical of laminated composites may be simulated.

Mass matrix

The mass matrix for the thin shell element is formed using a lumped mass methodology. The total mass for the element is distributed evenly among the nodes and assigned to the displacement degrees of freedom. No values of rotary inertia are assigned to the rotation degrees of freedom.

Geometric stiffness matrices

The effect of in-plane stresses on the bending stiffnesses of an element is handled through the calculation of geometric stiffness matrices. Then, for initially stressed structures, or for analysis of structures subject to geometric non-linearities, the geometric stiffness matrices are scaled with the stress resultants and added to the element's stiffness matrix to create a "stressed element" stiffness matrix.

In calculating the geometric stiffness matrices, the program uses a linear interpolation for the normal displacement. Although this is a lower order of approximation than that used for the element stiffness matrix, this is consistent in an energy sense.

Auxiliary Elements

SAPNEW provides a three-dimensional beam element with twelve degrees of freedom and a two degree of freedom linear linear spring element as auxiliary elements. The intended use of these elements is for modeling elastic supports for the structure (e.g. to include the effects of an elastic rotor disk in a turbine blade analysis). Thus, these elements have not been optimized for concurrency and vectorization beyond automatic compiler optimizations and their use should be limited.

Centrifugal forces

SAPNEW calculates the effective load due to constant rotation using the lumped mass matrix previously described.

Multi-Point Constraints

In addition to fixed single-point constraints, SAPNEW allows constraints wherein one degree of freedom is determined by a linear combination of up to four other degrees of freedom. This allows semi-fixed supports, as well as rigid members to be modeled. Note that the degrees of freedom, upon which a multi-point constrained degree of freedom depends, may not themselves be multi-point constrained.

3. PARALLELIZATION OF SAPNEW

Because of the tremendous computational effort involved in performing an aeroelastic analysis on a bladed disk assembly, improvements in program performance are very important. Parallel and/or vector processing seems to provide the best hope for improved computing speed. For this reason, SAPNEW was intended for use on a parallel processing computer (e.g. the Alliant FX/80). Several aspects of the program were designed for improved parallel efficiency.

Element Generation

During the element generation phase, the program calculates the element stiffness matrices and element mass matrices. These calculations are independent and thus, are well suited to concurrent execution. SAPNEW does perform all shell element calculations in parallel.

Linear Equation Solution

Crout decomposition (LDL^T) or Cholesky decomposition (LL^T) (for positive definite systems) are well known direct methods for the solution of a linear system. These algorithms are popular partly because they can take advantage of a compact "skyline" storage scheme for the stiffness matrix, although there can be substantial fill-in below the skyline.

These methods were designed for sequential operation. However, careful examination of the algorithms shows that there are operations which can be performed concurrently. The LL^T algorithm is given in Figure 1.

For i = 1 to n

$$L_{ii} = \sqrt{K_{ii} - \sum_{k=1}^{i-1} L_{ik}^2}$$

For j=i+1 to n

$$L_{ji} = \frac{K_{ji} - \sum_{k=1}^{i-1} L_{ik} L_{jk}}{L_{ii}}$$

Next j

Next i

Figure 1. Cholesky decomposition algorithm.

The calculations in the inner loop (j-loop) in the LL^T algorithm are independent of each other. Thus, this loop can be executed concurrently. Note, however, that the number of tasks to be performed in this loop changes with i. As i gets close to n, there are fewer tasks to perform, and consequently, there is little benefit from parallelization at this point. This fact limits the parallel efficiency that this algorithm can achieve.

After the matrix is factored, the solution is obtained by first forward substituting to solve $[L]\{y\} = \{F\}$ and then back-substituting to solve $[L]^T\{q\} = \{y\}$. These substitutions are inherently sequential operations and further limit the application of parallel processing to this algorithm. Thus, it is desirable to explore alternate algorithms on parallel machines.

Element-by-element preconditioned conjugate gradient (EBE-PCG) algorithms have been advocated for use in parallel/vector environments as being superior to the LDL^T decomposition algorithm. The conjugate gradient algorithm involves generating a set of mutually conjugate direction vectors. The quadratic total potential energy function is then minimized successively along each direction. Using exact arithmetic, it can be shown[7] that this algorithm will require at most n iterations to find the solution for an n degree of freedom problem. This property makes the conjugate gradient algorithm attractive among iterative methods. A version of the conjugate gradient algorithm which exploits the inherent element-level parallelism of a finite element model has been proposed by Law[8].

Further improvements in the performance of the conjugate gradient algorithm can be achieved through preconditioning. Preconditioning consists of transforming the stiffness matrix with an approximation of its inverse. This approximation can be as simple as a diagonal matrix[9], or much more sophisticated, such as the element-by-element preconditioner proposed by Hughes.[10]

The element by element conjugate gradient algorithm has proven to be relatively efficient in taking advantage of a parallel computing environment. However, its cost effectiveness is highly problem dependent. For finite element problems which generate a stiffness matrix with a large mean bandwidth, the EBE-PCG is the method of choice. For problems with low mean bandwidths, or involving multiple load cases it was found that the EBE-PCG cannot match the performance of the LL^T decomposition algorithm[11].

Thus, the SAPNEW program can use either a parallelized LL^T algorithm or the EBE-PCG algorithm to solve the linear systems that it generates. However, for blade models (which are generally very ill-conditioned) the EBE-PCG method may fail due to machine round-off, and it is recommended that the decomposition algorithm be used.

Eigen Analysis

To calculate the eigenvalues and eigenvectors, SAPNEW uses the subspace iteration procedure. This procedure involves projecting the stiffness and mass matrices on a desired subspace. This process is, in fact, parallelizable over the dimension of the subspace. SAPNEW calculates the projected mass and stiffness matrices in parallel.

4. EVALUATION OF SAPNEW

Validation

To check the accuracy of the SAPNEW program, several static and dynamic analyses of rectangular plates were carried out for various aspect ratios and mesh-sizes.

Descriptions of models are listed in Table 1. The results of the static analysis are listed in Table 2. The results of the dynamic analysis are listed in Table 3. Finally, the results of the dynamic restart analysis are listed in Table 4.

Table 1. Description of models

Model no	1	2	3	4	5	6	7	8
Aspect ratio (b/a)	1.0	1.0	1.0	1.0	1.4	1.4	1.4	1.4
Mesh size	10x10	20x20	30x30	50x50	10x10	20x20	30x30	50x50
Total D.O.F	287	1167	2649	7409	287	1167	2649	7409
Mean bandwidth	30	61	96	156	30	61	96	156

Notes: boundary condition : simple supports on all four sides
plate length : a = 20.0 m
bending rigidity : 0.08333 N-m
mass density : 0.0001 kg
loading type
- Concentrated load applied at mid-point of plate. (F = 1.0 N)
- Uniform pressure load (p = 0.1 N/m^2)

Table 2. The results of static analysis

Aspect ratio of shell element	Loading type	Mesh size	Maximum deflection (mm)	theory (mm)	relative error (%)
1.0	F	10x10	55.007	55.903	1.60
		20x20	55.484		0.74
		30x30	55.623		0.50
		50x50	55.847		0.10
	p	10x10	764.31	782.65	2.34
		20x20	776.04		0.84
		30x30	779.51		0.41
		50x50	781.08		0.11
1.4	F	10x10	70.329	71.518	1.66
		20x20	71.050		0.65
		30x30	71.303		0.31
		50x50	71.374		0.20
	p	10x10	1333.4	1359.04	1.88
		20x20	1353.5		0.41
		30x30	1361.1		0.15
		50x50	1358.9		0.10

Notes: F : concentrated load at the mid-point of plate
p : uniform pressure load

Table 3. The results of the dynamic analysis

Model no.		Frequencies of modes (Hz)						
		1	2	3	4	5	6	7
1	C	4.5717	11.331	11.331	18.216	22.776	22.776	29.777
	T	4.5048	11.262	11.262	18.019	22.524	22.524	29.281
	E	1.5	0.6	0.6	1.1	1.1	1.1	1.7
2	C	4.5079	11.279	11.279	18.069	22.587	22.587	29.406
	T	4.5048	11.262	11.262	18.019	22.524	22.524	29.281
	E	0.06	0.15	0.15	0.28	0.27	0.27	0.4
3	C	4.5061	11.269	11.269	18.041	22.551	22.551	29.336
	T	4.5048	11.262	11.262	18.019	22.524	22.524	29.281
	E	0.02	0.06	0.06	0.12	0.1	0.1	0.18
4	C	4.5053	11.264	11.264	18.027	22.534	22.534	29.301
	T	4.5048	11.262	11.262	18.019	22.524	22.524	29.281
	E	0.01	0.02	0.02	0.04	0.04	0.04	0.68
5	C	3.4594	6.9313	10.291	13.208	19.564	20.845	27.752
	T	3.4016	6.8492	10.159	13.065	19.352	20.639	27.396
	E	1.7	1.2	1.3	1.1	1.1	1.0	1.3
6	C	3.4458	6.9176	10.230	13.143	19.352	20.701	27.451
	T	3.4016	6.8492	10.159	13.065	19.352	20.639	27.396
	E	1.3	1.0	0.7	0.6	0.8	0.3	0.2
7	C	3.4390	6.9245	10.230	13.104	19.448	20.680	27.451
	T	3.4016	6.8492	10.159	13.065	19.352	20.639	27.396
	E	1.1	1.1	0.7	0.3	0.5	0.2	0.2
8	C	3.4322	6.8971	10.169	13.130	19.390	20.680	27.478
	T	3.4016	6.8492	10.159	13.065	19.352	20.639	27.396
	E	0.9	0.7	0.1	0.5	0.2	0.2	0.3

Notes: C : calculated value
T : theoretical value (from reference [12])
E : relative error (%)

Test models

The models used for evaluating the SAPNEW program were typical propfan blades: SR5 and SR7L. The NTOS conversion program was used to convert a NASTRAN models of these blades to the SAPNEW data input format.

Figure 2. shows the geometry of the SR5 blade. Table 4. lists the statistics for this blade model. The SR5 test case consisted of determining the three lowest eigenvalues and their corresponding mode shapes using geometric stiffness generated by the static solution of the blade loaded by centrifugal forces. The SR5 blade model constructed using homegeneous and isotropic material properties.

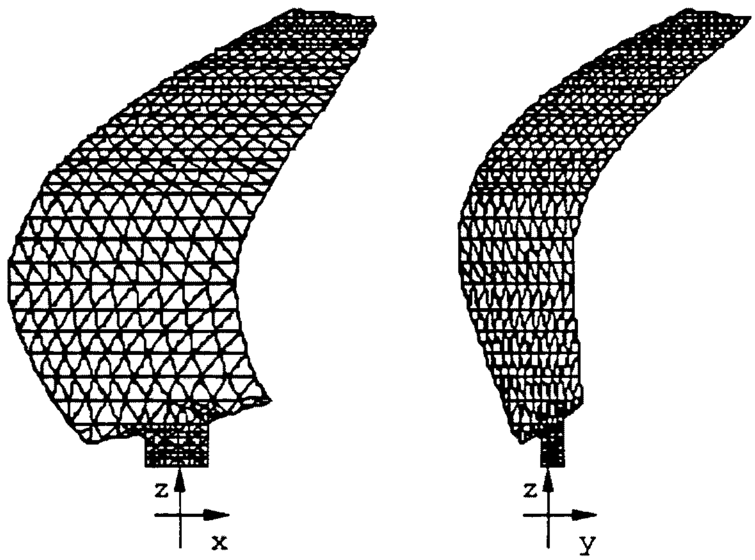


Figure 2. SR5 blade geometry.

Table 4. SR5 blade model statistics.

General:	
Types of elements	Triangular Thin Shell
Number of elements	702
Number of nodes	402
Number of degrees of freedom	2360
Stiffness Matrix:	
Number of working elements	321117
Maximum half-bandwidth	2008
Mean half-bandwidth	136

Figure 3. shows the geometry of the SR7L blade. Table 5. lists the statistics for this blade model. The SR7L test case consisted of determining the six lowest eigenvalues and their corresponding mode shapes using geometric stiffness generated by the static solution of the blade loaded by centrifugal forces. The SR7L blade model was constructed using material properties derived from classical plate analysis of laminated composite structures.

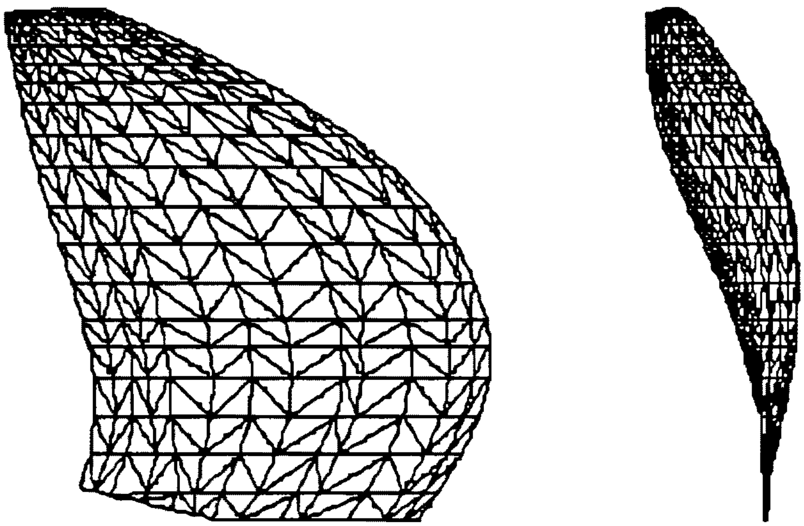


Figure 3. SR7L blade geometry.

Table 5. SR7L blade model statistics.

General:	
Types of elements	Triangular Thin Shell
Number of elements	449
Number of nodes	267
Number of degrees of freedom	1550
Stiffness Matrix:	
Number of working elements	208793
Maximum half-bandwidth	1474
Mean half-bandwidth	134

Results

The calculated natural frequencies for both blade models are given in Table 6. This table also presents the frequencies calculated by MSC/NASTRAN for comparison. The lowest mode frequency discrepancy between SAPNEW and MSC/NASTRAN is due to differences in the manner in which geometric stiffening is accounted for. For the geometric stiffness calculations, NASTRAN uses the same interpolation functions for normal displacements as were used in the bending stiffness calculations. SAPNEW uses a linear interpolation for the normal displacement. Although this is a lower order of approximation than that used for the element stiffness matrix, this is consistent in an energy sense.

Table 6. Blade model results.

(a.) SR5

Mode	Frequency (Hz) SAPNEW	Frequency (Hz) MSC/NASTRAN	Relative error (%)
1	174.60	151.32	15.38
2	287.41	281.11	2.24
3	563.16	586.33	-3.95

(b.) SR7L

Mode	Frequency (Hz) SAPNEW	Frequency (Hz) MSC/NASTRAN	Relative error (%)
1	51.34	43.52	17.98
2	90.50	94.40	-4.14
3	105.91	108.50	-2.39
4	149.82	147.08	1.87
5	175.52	182.47	-3.80
6	245.05	231.25	5.97

The times required by the SAPNEW program to run the test cases on the Alliant FX/80 for different code optimization options are given in Table 7. The corresponding speed-up values are listed in Table 8. and presented in Figure 4.

Table 7. Time results (All times in sec.).

	Number of Processors					
	1	2	3	4	5	6
Without Vectorization						
SR5	190.27	106.45	78.22	73.67	72.09	53.55
SR7L	233.44	124.73	88.56	71.92	70.21	54.69
With Vectorization						
SR5	105.26	63.31	50.31	47.24	46.28	41.05
SR7L	105.45	61.09	47.25	41.56	41.12	38.58

Table 8. Speedup results.

	Number of processors					
	1	2	3	4	5	6
Eigen Analysis only						
SR5	1.00	1.84	2.44	2.55	2.52	3.12
SR7L	1.00	1.89	2.59	3.04	3.01	3.31
Total Problem Run						
SR5	1.00	1.66	2.09	2.23	2.27	2.56
SR7L	1.00	1.73	2.23	2.54	2.56	2.73

Note : Total problem run includes: input, element formulation, static analysis, eigen analysis, and output.

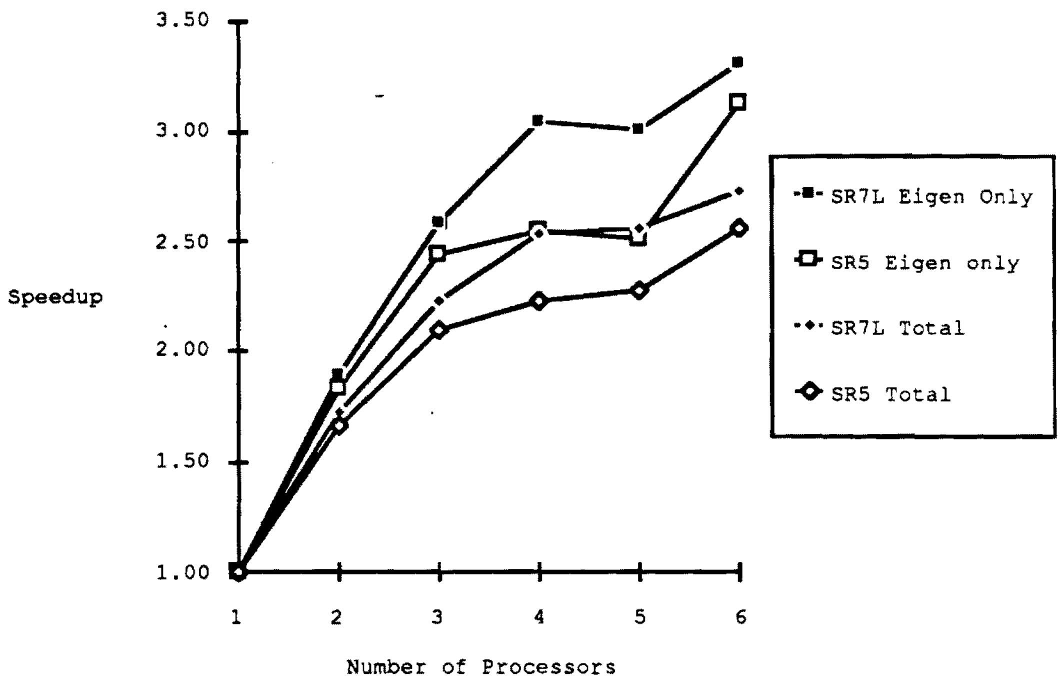


Figure 4. Speedup results.

The dips in the curves for the eigen analysis speedup are cause by the fact the there are six tasks for the SR5 test model and twelve tasks for the SR7L test model which are performed concurrently. The number of tasks is related to the number of modes to be found.

5. REFERENCES

1. Kaza, K.R.V. and Kielb, R.E. , " Flutter and Response of Mistuned Cascade in Incompressible Flow", *AIAA Journal*, vol. **20**, no.2, pp.1120-1127, 1982.
2. Bathe, K.J., Wilson, E.L., and Peterson, F.E., "SAP IV-- A Structural Analysis Program for Static and Dynamic Analysis of Linear Structural Systems", *EERC Report No. 73-11*, College of Engineering, University of California, Berkeley, June 1973.
3. Bathe, K.J., Ozdemir, H., and Wilson, E.L., "Static and Dynamic Geometric and Material Nonlinear Analysis", *SESM Report No. 74-4*, College of Engineering, University of California, Berkeley, February 1974.
4. Bathe, K.J., *Finite element procedures in engineering analysis*, Prentice-Hall company, pp.672-684, 1982.
5. Narayanan, G.V. and Kaza, K.R.V., "ASTROP2 Users' Manual: A Program for Aeroelastic Stability Analysis of Propfans", NASA TM 4304, August 1991.
6. Clough R.W. and Felippa, C.A.," A Refined Quadrilateral element for analysis of plate bending", *Proceedings 2'nd conference on matrix methods in structural mechanics*, Wright Peterson AFB, Ohio, 1968.
7. Hestenes, M.R., *Conjugate Direction Methods in Optimization*, Springer-Verlag, pp. 116-125, 1980
8. Law, K.H.,"Parallel Finite Element Solution Method", *Computer & Structures* vol.**23**, No.6, pp845-858, 1986.
9. Argyris, J. et al., "Τα Παντα Ρει", *Computer Methods in Applied Mechanics and Engineering*, vol.**51**, pp.289-365, 1985.
10. Hughes, T.J.R., et al. , "New Alternating Direction procedures in Finite Element Analysis based upon eBe Approximate Factorizations", *Computer Methods for Nonlinear solids and Structural Mechanics*, AMD vol.**54**, pp.75-109, 1983.
11. Watson, B.C. and Kamat, M.P., "A Study of Equation Solvers for Linear and Non-Linear Finite Element Analysis on Parallel Processing Computers", *Proceedings 33rd AIAA/ASME/ ASCE/AHS SDM Conference, Paper # 92-2478, Dallas, TX, April 1992*, .
12. Timoshenko S.P., Woinowsky-Krieger S. ,*Theory of plates and shells*, McGraw-Hill company, pp.140-158, 1959.

APPENDIX I. USER'S GUIDE FOR SAPNEW

File names

Executable file

The executable file is located on the Alliant FX/80 at NASA Lewis Research Center. The program name is **sapnew**. The program synopsis is as follows:

```
$ sapnew [-e|c|n] infln
```

The input file should be named *infln*.dat where *infln* is a user chosen file name prefix. The program will write its output into a file named *infln*.out .

-e This option will cause the program to use the element-by-element conjugate gradient algorithm to solve the linear system for static analysis. If the data file specifies dynamic analysis, this option has no effect. If the model has multi-point constraints, this option should not be used.

-c This option will cause the program to use the conjugate gradient algorithm on the assembled stiffness matrix to solve the linear system for static analysis. If the data file specifies dynamic analysis, this option has no effect.

-n This option causes the program to generate a data file for the ASTROP aeroelastic analysis program. This data will be written to a file named *infln*.nasty . If the input data specifies static analysis, this flag has no effect.

Source files

The source files are written in Alliant's FX/Fortran. This is an extension of Fortran/77 with directives to specify parallelization and vectorization. These directives appear as comments to standard Fortran. They are located on the Alliant FX/80 together with an associated Makefile. A short description of each module follows:

sapmain.f : main program code.
sapsubs.f : general subroutines.
saprecur.f : code to generate the shell element stiffness and mass matrices.
sapsolv.f : code for Cholesky decomposition of stiffness matrix
sapdyn.f : code for eigen analysis
sapecgm.f : code for element-by-element conjugate gradient algorithm
sapcgm.f : code for general conjugate gradient algorithm

Auxiliary files

Auxiliary files may be created by the program (at the user's option) for the possibility of restarting a dynamic analysis to calculate more eigen values/vectors.

modal.inf :	storage of modal information
stif.inf :	storage of assembled stiffness matrix
mass.inf :	storage of assembled mass matrix and the LM array

Sample data files

Sample data files for static and modal analysis of propfan blades (SR5 and SR7L) are available on the Alliant FX/80.

sr5.dat :	static analysis of an isotropic blade with centrifugal load
sr5dyn2.dat:	modal analysis of an isotropic blade with geometric stiffening due to centrifugal load.
sr7l.dat:	static analysis of a composite blade with centrifugal load. This model uses beam and spring elements to simulate an elastic support.
sr7ldyn2.dat:	modal analysis of a composite blade with geometric stiffening due to centrifugal load.

Input data file format

Static analysis

Title card	(section 1)
Control information card	(section 2)
Node information cards	(section 4)
Concentrated load information cards	(section 5)
Element information cards	(section 7)
Centrifugal load information cards	(section 8)
Load factor cards	(section 9)

Modal analysis

Without *geometric stiffening*

Title card	(section 1)
Control information card	(section 2)
Dynamic control information card	(section 3)
Node information cards	(section 4)
Concentrated mass information cards	(section 6)
Element information cards	(section 7)

With *geometric stiffening*

Title card	(section 1)
Control information card	(section 2)
Dynamic control information card	(section 3)
Node information cards	(section 4)
Concentrated load information cards	(section 5)
Element information cards	(section 7)
Centrifugal load information cards	(section 8)

Restarting the eigen value/vector analysis

Title card	(section 1)
Control information card	(section 2)
Dynamic control information card	(section 3)

1. Title card

Format	Description
A80	Title of analysis

2. Control information card

<u>Format</u>	<u>Description</u>
I5	Analysis code

0; Static analysis

>0; Eigen analysis
Analysis code -1 = number of static solution iterations for geometric stiffness computation
(E.g. Analysis code = 1 means eigen analysis with no geometric stiffening effect accounted for.
Analysis code = 2 means eigen analysis with one static analysis to compute geometric sitffness matrices.
Analysis code = 3 means eigen analysis with two static analysis iterations to compute geometric stiffness matrices .
etc.)

I5 Number of node points

I5 Number of element groups

I5 Number of load cases or modes

Analysis code = 0; Load cases (not including centrifugal load)
Analysis code >0; Modes

I5 Flag for execution mode

0; Execute
1; Input data verification

I5 Flag for centrifugal load

0; No centrifugal loads
1; Use centrifugal loads

Note: If analysis code > 1 and centrifugal loading is not used, then one load case (with concentrated loads) is expected.

3. Dynamic control information card

Format	Description
F10.0	Cut-off frequency
	Default = 1.0×10^{9}
F10.0	Error tolerance in the subspace iteration procedure
	Default = 1.0×10^{-6}
I5	Maximum number of iterations
	Default = 16
I5	Flag for shifting
	0 ; Do not use shifting 1 ; Use shifting
F10.0	Shifting factor
I5	Flag for Sturm sequence check
	0 ; Do not check 1 ; Check
I5	Flag for printing the iteration procedure
	0 ; Do not print 1 ; Print
I5	Flag for restart execution
	0 ; Initial execution -1 ; Restart execution
I5	Flag for saving modal parameters
	0 ; Do not save 1 ; Save for the later usage

Notes: 1. Normally, the lowest eigenvalues are computed. Shifting can be used to find the closest eigenvalues to the specified shifting factor.
2. The Sturm sequence check can be used to insure that the desired eigenvalues were in fact the ones that were found.

4. Node information cards

Node information cards (one for each node)

Format	Description
I5	Node number
6I5	Boundary condition code for X, Y, Z, RX, RY, RZ directions 0; Free 1; Fixed >1; Constrained by Multi-Point-Constraint
F10.0	X-coordinate
F10.0	Y-coordinate
F10.0	Z-coordinate
I5	Node generation code

Note: Node generation may be used if some nodes are evenly spaced along some line segment. The node generation code is the increment in node number to be used for the generated nodes. For example, these input cards:

```
    8    0    0    0    0    0    0       0.0       0.0       0.0    2
   18    0    0    1    1    1    1      20.0       0.0      25.0    0
```

would generate the following nodes:

```
    8    0    0    0    0    0    0       0.0       0.0       0.0
   10    0    0    0    0    0    0       4.0       0.0       5.0
   12    0    0    0    0    0    0       8.0       0.0      10.0
   14    0    0    0    0    0    0      12.0       0.0      15.0
   16    0    0    0    0    0    0      16.0       0.0      20.0
   18    0    0    1    1    1    1      20.0       0.0      25.0
```

Note that the node number increment (Node generation code) is specified on the first card of this input pair.

Following all node information cards:

- *Multi-point constraint information cards (one for each multi-point constrained DOF)*

<u>Format</u>	<u>Description</u>		
I5	Node 1	}	DOF 1
I5	Direction 1=X, 2=Y, ..., 6=RZ		
F10.0	Coefficient 1	}	TR 1
I5	Node 2	}	DOF 2
I5	Direction 1=X, 2=Y, ..., 6=RZ		
F10.0	Coefficient 2	}	TR 2
I5	Node 3	}	DOF 3
I5	Direction 1=X, 2=Y, ..., 6=RZ		
F10.0	Coefficient 3	}	TR 3
I5	Node 4	}	DOF 4
I5	Direction 1=X, 2=Y, ..., 6=RZ		
F10.0	Coefficient 4	}	TR 4

Note: The constraint is formed as:

Constrained DOF = TR1*DOF1 + TR2*DOF2 + TR3*DOF3 + TR4*DOF4

5. Concentrated load information cards

(one set for each load case)

Load control card

Format	Description
I5	Load case number
I5	Number of loads in this load case

Concentrated load cards (one for each load)

Format	Description
I5	Node number at which the load is applied
I5	Code for the direction of the applied load
	1=X, 2=Y, ..., 6=RZ
F10.0	Magnitude of the applied load

6. Concentrated mass information cards

(one for each concentrated mass)

Format	Description
I5	Node number
F10.0	Mass in the x-dir.
F10.0	Mass in the y-dir.
F10.0	Mass in the z-dir.
F10.0	Inertia in the rx-dir.
F10.0	Inertia in the ry-dir.
F10.0	Inertia in the rz-dir.

Note: A blank card signals the end of the concentrated mass input data. Thus, even for no concentrated masses, a blank card must be present (for dynamic analysis).

7. Element information cards

Shell element control card

Format	Description
I5	Code for the element type
	1 ; shell element
I5	Number of shell elements
I5	Number of shell material property sets

Shell material property cards *(a pair of cards for each shell material property set)*

Format	Description
I5	Material property number
20X	
F10.0	Mass density
F10.0	Thermal expansion coefficient in the x-dir.
F10.0	Thermal expansion coefficient in the y-dir.
F10.0	Thermal expansion coefficient in the z-dir.

Format	Description
F10.0	C_{11} of the material coefficient matrix $[C_{ij}]$
F10.0	C_{12} of the material coefficient matrix $[C_{ij}]$
F10.0	C_{13} of the material coefficient matrix $[C_{ij}]$
F10.0	C_{22} of the material coefficient matrix $[C_{ij}]$
F10.0	C_{23} of the material coefficient matrix $[C_{ij}]$
F10.0	C_{33} of the material coefficient matrix $[C_{ij}]$

Note: The material coefficient matrix $[C_{ij}]$ should be as follows:

For isotropic materials:
Plane stress:

$$[C_{ij}] = \frac{E}{1-\nu^2}\begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix}$$

Plane strain:

$$[C_{ij}] = \frac{E}{(1+\nu)(1-2\nu)}\begin{bmatrix} 1-\nu & \nu & 0 \\ \nu & 1-\nu & 0 \\ 0 & 0 & \frac{1-2\nu}{2} \end{bmatrix}$$

For orthotropic materials:
Plane stress:

$$[C_{ij}] = \frac{E_y}{1-n\nu_y{}^2}\begin{bmatrix} n & n\nu_y & 0 \\ n\nu_y & 1 & 0 \\ 0 & 0 & m(1-\nu_y{}^2) \end{bmatrix}$$

Plane strain:

$$[C_{ij}] = \frac{E_y}{(1+n\nu_y)(1-2n\nu_y)}\begin{bmatrix} 1-n\nu_y & n\nu_y & 0 \\ n\nu_y & 1-n\nu_y & 0 \\ 0 & 0 & \frac{m(1-n\nu_y)}{2} \end{bmatrix}$$

where
E : Young's modulus
G : shear modulus
ν : Poisson's ratio
n : E_x / E_y
m : G_x / G_y

Shell element load multiplier cards (5 cards)

Format	Description
4F10.0	pressure load multiplier factors

Format	description
4F10.0	thermal load multiplier factors

Format	description
4F10.0	x-acceleration multiplier factors

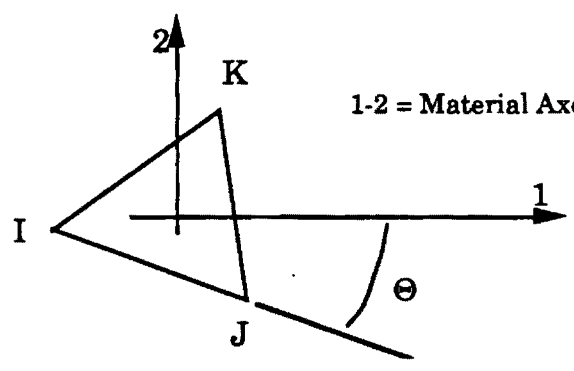
Format	description
4F10.0	y-acceleration multiplier factors

Format	description
4F10.0	z-acceleration multiplier factors

Note: The four multipliers for these loads form four different loading conditions. Within each loading condition, these values determine the relative amount of each load type (e.g. pressure to thermal loading). For each problem load case, these four loading conditions will be scaled (through a load factor card [section 9]) and superposed and then added to the load vector.

Shell element description card (one card for each shell element)

Format	Description
I5	Shell element number
I5	Node I
I5	Node J
I5	Node K
I5	Node L
I5	Mid-point node
I5	In-plane material property number
I5	Bending material property number
I5	Element generation code (See note 5. on next page)
F7.0	Thickness of the element
F7.0	Transverse pressure on the element
F7.0	Temperature of the element
F7.0	Temperature gradient accross the thickness of the element
F7.0	Theta (See Figure below)



Notes: 1. The elements must be consecutively numbered, and input in order.

2. If the element is triangular, node L and the mid-point node should be zero.

3. If the element is quadrilateral and the behavior at the mid-point needs to be known, the mid-point node should be specified. Otherwise, set this node to zero.

4. If the material is isotropic or the element is quadrilateral, then theta should be greater than 180.

5. Different in-plane and bending material properties are allowed so that laminated composite materials may be simulated. (This is similar to NASTRAN. However, unlike NASTRAN, this shell element does not include the transverse shear deformation.)

6. Automatic element geneneration can be used if the relative node numbers for some elements remain constant.

For example, the following input cards:

```
16    1    3    4    2    0    1    1    0    0.1    0.0    0.0    0.0  200.0
20    9   11   12   10    0    1    1    2    0.1    0.0    0.0    0.0  200.0
```

would generate the following elements:

```
16    1    3    4    2    0    1    1         0.1    0.0    0.0    0.0  200.0
17    3    5    6    4    0    1    1         0.1    0.0    0.0    0.0  200.0
18    5    7    8    6    0    1    1         0.1    0.0    0.0    0.0  200.0
19    7    9   10    8    0    1    1         0.1    0.0    0.0    0.0  200.0
20    9   11   12   10    0    1    1         0.1    0.0    0.0    0.0  200.0
```

Note that the node increment (element generation code) is specified on the second card in this pair.

Beam element control card

Format	Description
I5	Code for the element type
	2 ; beam element
I5	Number of beam elements
I5	Number of beam geometric property sets
I5	Number of beam fixed-end force sets
I5	Number of beam material property sets

Beam material property cards *(one card for each beam material property set)*

Format	Description
I5	Beam material property set number
F10.0	Young's modulus
F10.0	Poisson's ratio
F10.0	Mass density
F10.0	Weight per unit length

Beam geometric property cards *(one card for each beam geometric property set)*

Format	Description
I5	Geometric property set number
F10.0	Axial cross section area
F10.0	Cross section area for shear 1
F10.0	Cross section area for shear 2
F10.0	Torsion coefficient 'J'
F10.0	Second area moment for axis 1
F10.0	Second area moment for axis 2

Beam element load multiplier cards (3 cards)

Format	Description
4F10.0	x-acceleration load multiplier

Format	Description
4F10.0	y-acceleration load multiplier

Format	Description
4F10.0	z-acceleration load multiplier

Beam fixed end force cards *(a pair of cards for each fixed-end force set)*

Format

I5

6F10.0

Format

F15.0

5F10.0

Beam element description cards *(one card for each beam element)*

Format	Description
I5	Element number
I5	Node I
I5	Node J
I5	Node K
I5	Material property set number
I5	Geometric property set number
4I5	End loads
I6	End code for node I
I6	End code for node J

Note: The beam axis connects nodes I & J. The vector from node I to node K detemines the cross section axis 1

Spring element control card

<u>Format</u>	<u>Description</u>
I5	Code for the element type
	3 ; spring element
I5	Number of spring elements

Spring element data card (one for each element)

<u>Format</u>	<u>Desription</u>
I5	Node I
I5	Node J
I5	Direction code
	1=X, 2=Y, ..., 6=RZ
F10.0	Spring stiffness

8. Centrifugal load information card (only if centrifugal loading is used)

Format	Description
F10.0	X-component of spin axis vector
F10.0	Y-component of spin axis vector
F10.0	Z-component of spin axis vector
F10.0	Spin rate in radians/second
F10.0	Unit conversion factor

Note: Spin axis passes through coordinate system origin.

9. Load factor card (one for each load case (**not** centrifugal loading))

Format	Description
4F10.0	Element load factors

APPENDIX II. NTOS - A CONVERSION UTILITY

To make SAPNEW more convenient to use, a conversion utility named NTOS (Nastran TO Sapnew) was written. This utility changes the format of a NASTRAN input data deck to that used by SAPNEW. NTOS is located on the Alliant FX/80 at NASA Lewis Research Center. The procedure for using NTOS on the Alliant is as follows:

```
$ ntos <nasdatafile >sapdatafile
```

where:

nasdatafile = NASTRAN input data filename

sapdatafile = SAPNEW input filename (must end in .dat)

The NTOS program only converts the BULK DATA section of the NASTRAN input data file. The user must manually edit the resulting SAPNEW file to include control information. (For example, the title card.) Following is a list of the NASTRAN bulk data cards which NTOS processes:

CBAR
CELAS1
CTRIA3
GRID
MAT1
MAT2
PBAR
PELAS
PSHELL

Any other cards in the bulk data deck will be ignored by NTOS. Thus the user must manually convert any other options. In particular, the user must manually add data cards for multi-point constraints, for centrifugal forces, and for any load cases that are desired.

The user must adjust the output of NTOS for either static or dynamic analysis. If dynamic analysis is desired, the dynamic control card must be entered manually (insert a blank line to accept control defaults).

E-16-1103

M. Vecon

GEORGIA INSTITUTE OF TECHNOLOGY

ATLANTA, GEORGIA 30332

SCHOOL OF
AEROSPACE ENGINEERING

404-894-3000

DANIEL GUGGENHEIM SCHOOL
OF AERONAUTICS

January 25, 1989

Dave Janetzke
NASA Lewis Research Center
Mail Stop 23/3
21000 Brookpark Rd.
Cleveland, OH 44135

Re: NASA Grant No. NAG-3-985

Dear Dave:

It was indeed very nice meeting you and Durbha at Lewis. We all enjoyed our visit with you.

Enclosed, please find a copy of our transparencies, a copy of the Shared Utility Subroutine Package on Flex 32 and a preliminary proposed budget page for grant renewal.

As we discussed, I hope this package will serve as the semi-annual progress report on the grant.

Thanks again for your time during our recent visit to Lewis.

Best regards,

Manohar P. Kamat

MPK/jr

Enclosure

cc: Office of Contract Administration, GA Tech, RE: E-16-A05.

Concurrent Processing Methods for Analysis and Design in Structural Mechanics

School of Aerospace Engineering
Georgia Institute of Technology
Atlanta, Georgia 30332

January 19, 1988.

NASA LEWIS RESEARCH CENTER
Cleveland, Ohio

Graduate Students Engaged in Concurrent Processing Research

- Andrew Kurdila (Ph.D. – Completed)
- Chris Ayers (M.S. – June '89)
- Chung-yul Song (Ph.D.)
- Greg Mascoli (M.S., Ph.D.?)
- Synn-youp Sang (M.S., Ph.D.)
- John Andersen (Ph.D.)

I. TASKS COMPLETED:

1. Linear Algebra Subroutine Package –
   Multiprocessor Version Operational
   on FLEX 32 e.g. QR Algorithm

2. Conjugate Gradient Algorithm for
   Solution of Linear System of
   Equations –
   Multiprocessor Version Operational
   On FLEX 32.

II. ONGOING TASKS:

1. Development of Linear FE Code with
   - a. Preconditioned conjugate Gradient (e.g. EBE-PCG)
   - b. LDL^T

   Equation Solvers;

2. Dynamic Analysis — substructuring using Component Mode Methods

III. PROPOSED TASKS:

- Concurrent Nonlinear FE Static & Dynamic Analysis using EBE-PCG Schemes.
- Concurrent Optimization Code
  - a. Sequential Unconstrained Optimization (Simultaneous Analysis and Design);
  - b. Generalized Optimality Criteria;
  - c. Simultaneous Analysis, Design & Control.
- Dynamic Analysis
  - a. Time-Space Finite Elements;
  - b. Substructuring etc.
- Uniquely Parallel Models and Algorithms

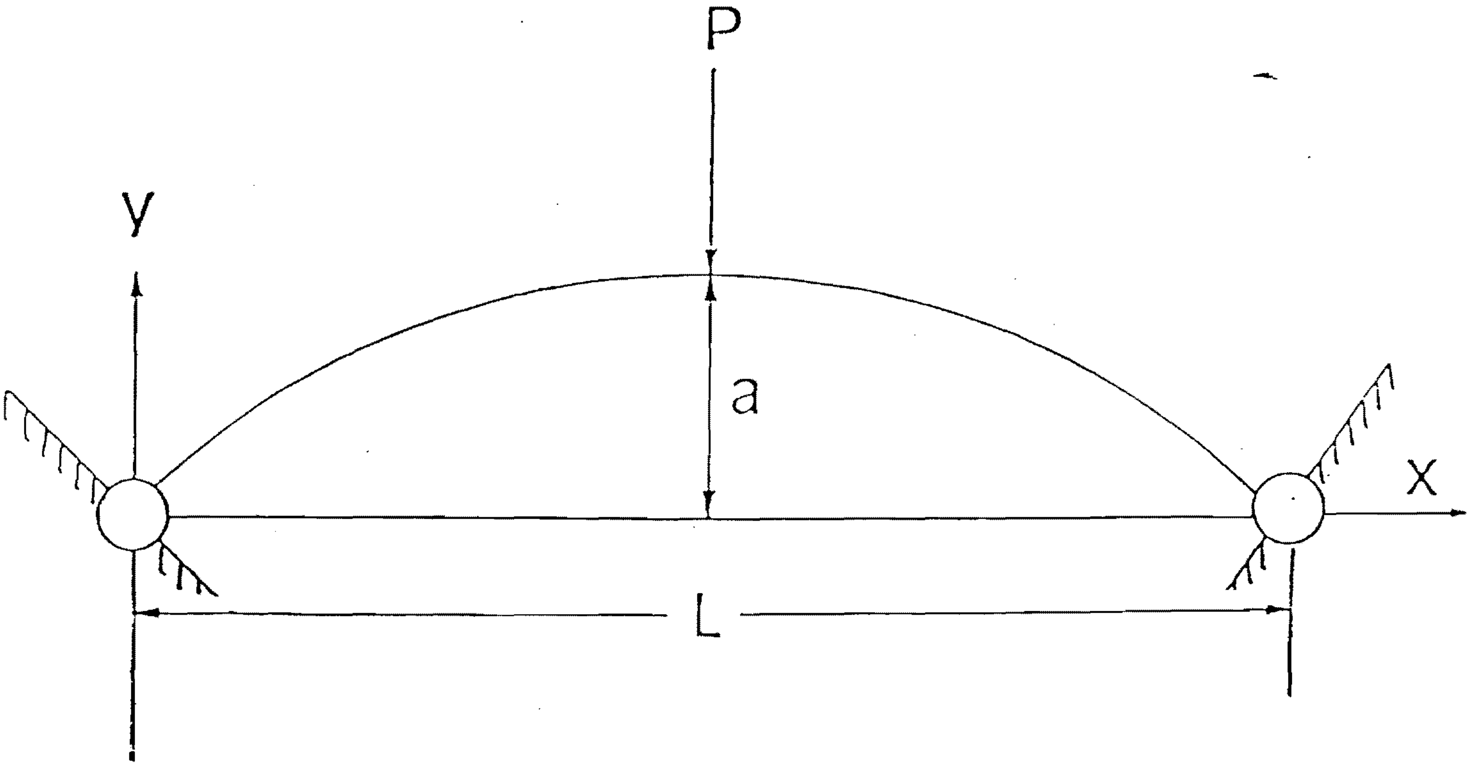
$$y = a \sin \frac{\pi x}{L}$$

$a = 5$ in., $L = 100$ in.

$A = 0.32\ in^2$, $I = 1\ in^4$,

$E = 10^7$ psi, 5-16 Noded Elements (1/2 span)

$(A = 24.5\ in^2,\ I = 76.5439\ in^4)$



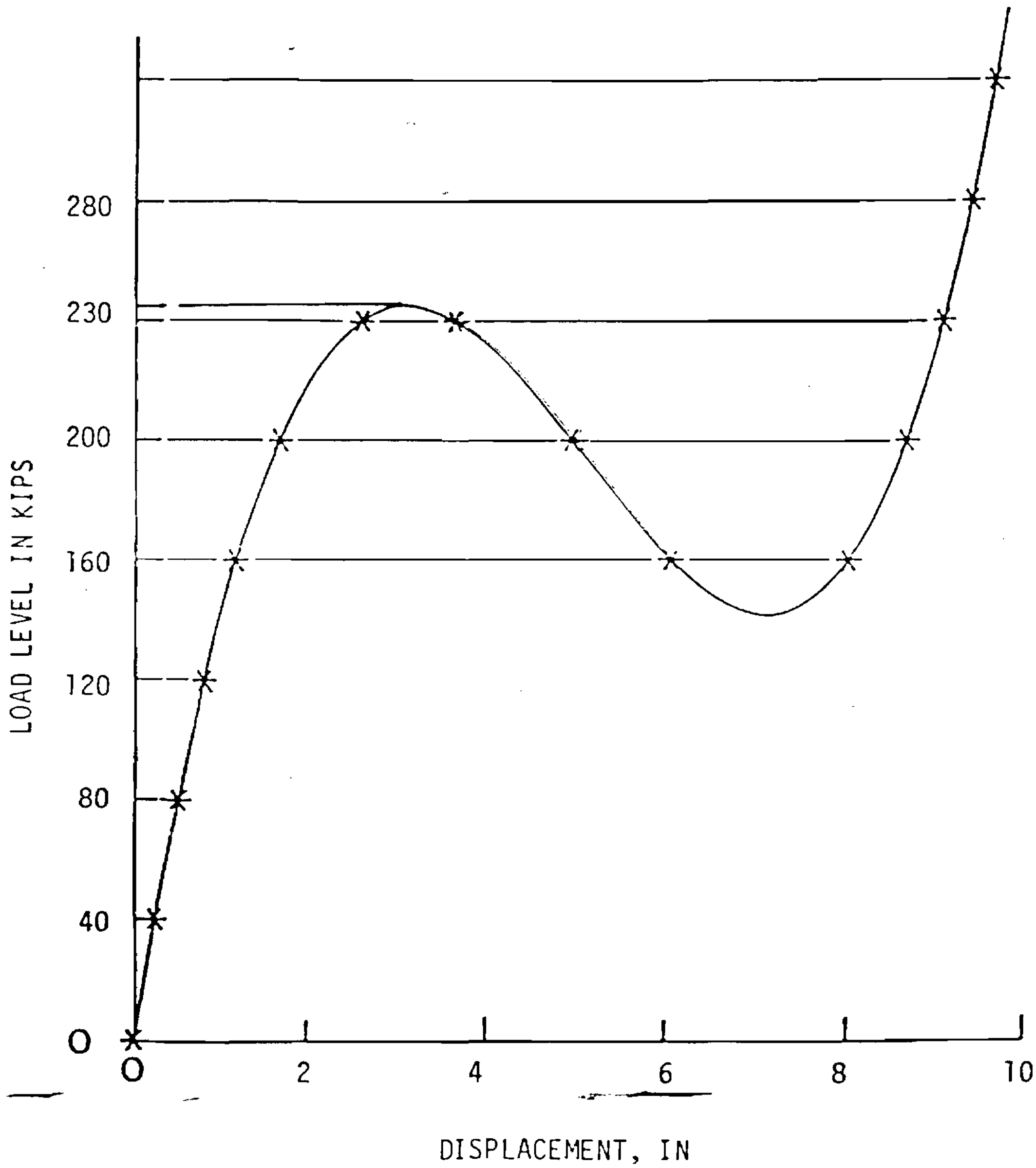
$P_{cr} = 3.064$ Kips

$= (230$ Kips$)$

Figure 2. Shallow Arch (130 degrees of freedom)

Table 1. Performance Evaluation of the Three Algorithms - One Load Step on the Thick Shallow Arch (Linear)

Algorithm	Normalized Parameters			
	Iterations	Function or Gradient Evaluations	CPU Time	Hessian Evaluations
CG	142 (1 restart)	143.5	17	0.0
EBE-PCG	37	36.5	5.78	1.0
QN	1.0	1.0	1.0	1.0



LOAD LEVEL IN KIPS
280
230
200
160
120
80
40
O
O
2
4
6
8
10
DISPLACEMENT, IN

OPTIMIZATION APPROACHES FOR STRUCTURES IN NONLINEAR RESPONSE

1. NESTED

Find X to Minimize W(X) subject to

$$g_j(X,U) \geq 0, \quad j = 1,2,...n$$

Such that $\quad$ F(U,P,X) $=$ 0

(Repeated Exact Nonlinear Analysis for every Iteration)

2. SIMULTANEOUS ANALYSIS AND DESIGN

Find (X,U) to Minimize W(X) subject to

$$g_j(X,U) \geq 0, \quad j = 1,2,...n$$

and the Equality Constraints

$$F(U,P,X) = 0$$

(Nonlinear Equilibrium Eqs. at each Iteration Solved Approx. and Sensitivity Derivatives much Easier to Calculate)

PENALTY FUNCTION FORMULATION FOR SIMULTANEOUS APPROACH

1. Mixed Extended Interior and Exterior Penalty Formulation

$$\phi(X, U, r) = cW(X) + r\sum_{j=1}^{n} p[g_j(X, U)] + \frac{c_1}{\sqrt{r}} F^T B^{-1} F$$

for r $= r_1, r_2 - \quad r_i \rightarrow 0$

B is a Preconditioner Matrix used to prevent ill-conditioning from a direct minimization of $F^T F$
Choice of B based on EBE Preconditioned Conjugate Gradient Method — Fully Parallel

2. One Dimensional Search $\phi(\lambda) = \min$ —Fully Parallel
   - Quadratic
   - Cubic
   - Golden Section Search

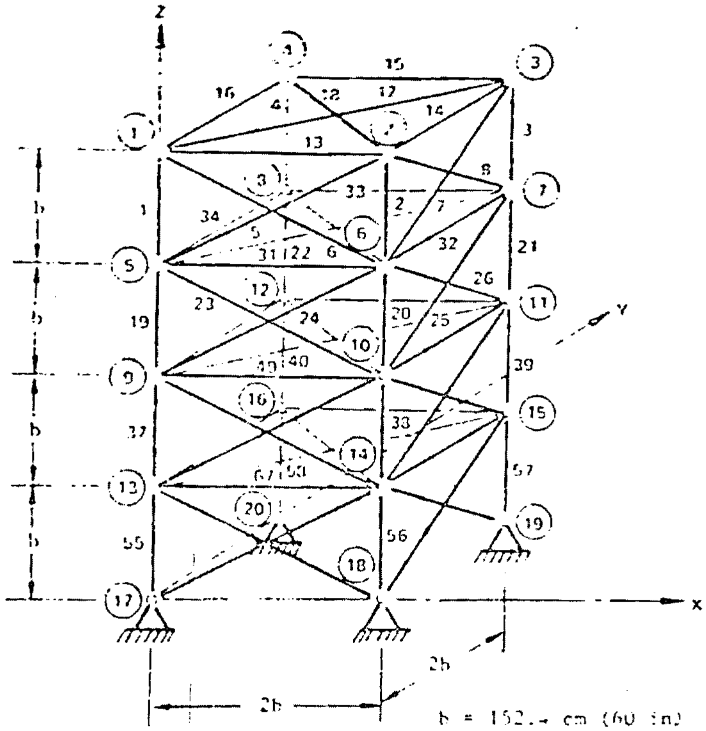
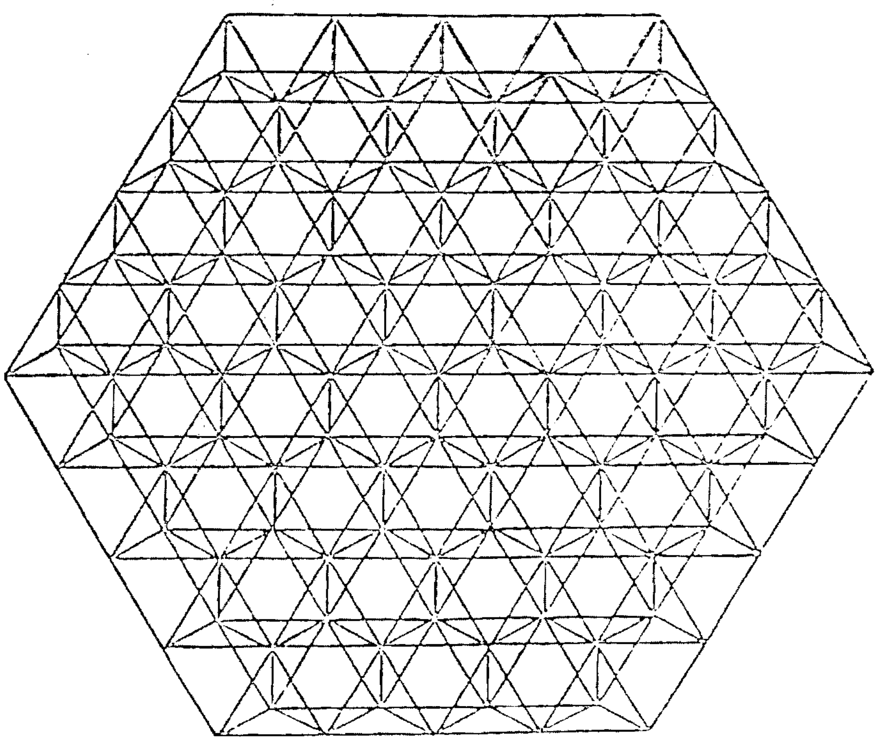


Figure 1. Seventy Two Bar Truss



PERFORMANCE OF THE PROPOSED ALGORITHM ON A SEQUENTIAL COMPUTING MACHINE

Table 1.a
Comparison of Results of Nested and Simultaneous Optimization Procedures for the 72-bar Truss (see Figure 1).

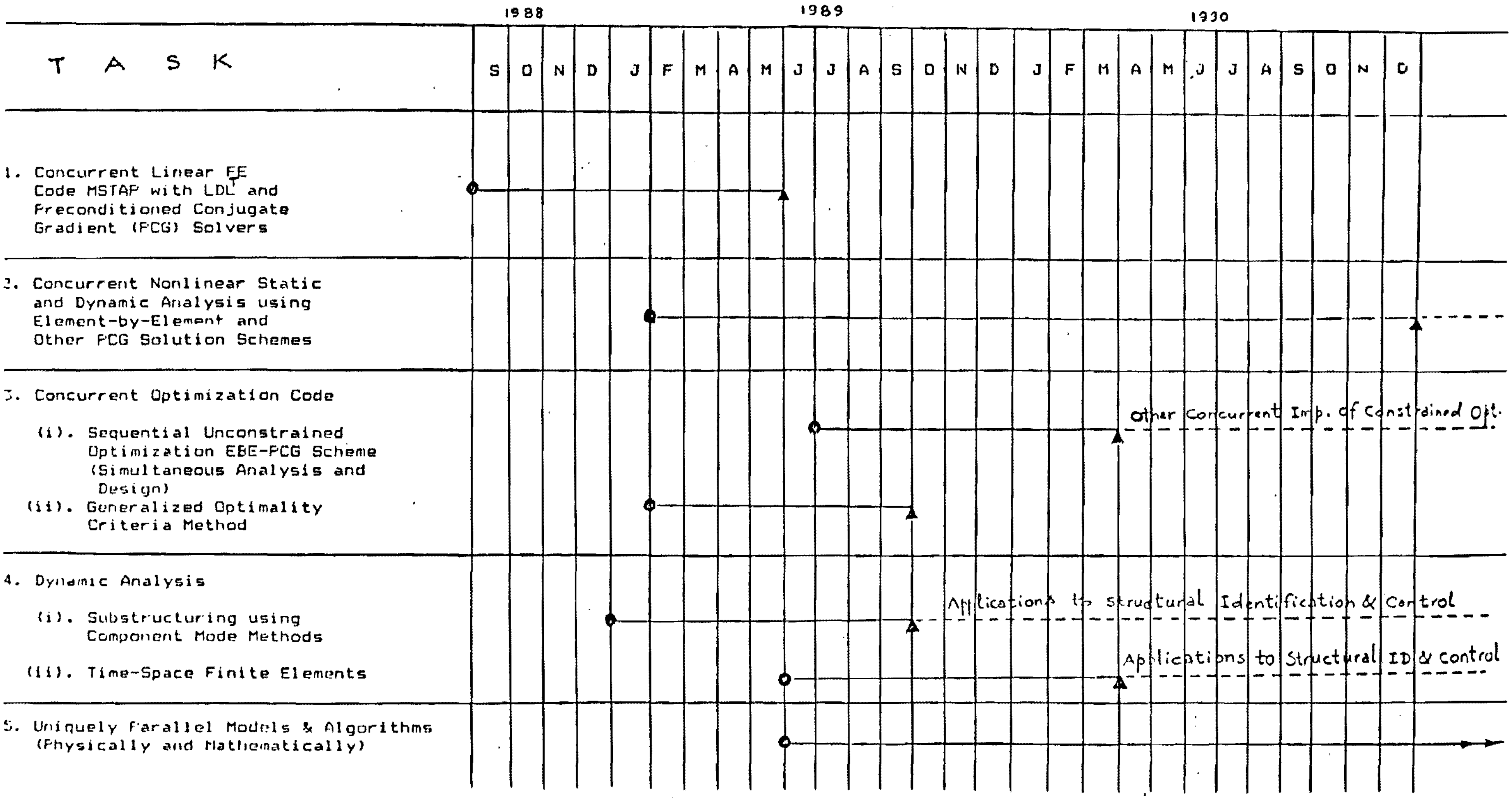
	Penalty-Function Formulation (Simultaneous)	Projected Langrangian (Nested)	
Mass (lb)	96.5	95.7	Nominal Loads
CPU (sec)	70.9 (20 r values, Eq. 5) 52.4 (10 r values, Eq. 5)	90.2	
Mass (lb)	493.7	498.9	Higher (x 10) Loads
CPU (sec)*	222.6 (20 r values, Eq. 5) 258.4 (10 r values, Eq. 5)	288.3	

Table 1.b
Comparison of Results of Nested and Simultaneous Optimization Procedures for the 420-bar Antenna (see Figure 2).

	Penalty-Function Formulation (Simultaneous)	Projected Langrangian (Nested)	Reduced Gradient (Nested)
Mass (lb)	1534	failed to converge after 96 hours of CPU	1519.6
CPU(hr)**	8.37		31.944

*IBM 3081
**IBM 4341

CONCURRENT PROCESSING METHODS FOR ANALYSIS AND DESIGN
IN STRUCTURAL MECHANICS



SHRUTL

SHARED UTILITIES LIBRARY*

Andy Kurdila

September 26, 1988

* Currently Available on FLEX 32

MODULE SUMMARY

SHARED MEMORY APPLICATIONS MODULES

*	DSHMGS	- "Double Precision Shared Memory Modified Gram Schmidt
*	DSHCQR	- "Double Precision Shared Memory Givens QR Decomposition"
*	DHLDSH	- "Double Precision Shared Memory Householder QR Decomp."
*	DSHOSV	- "Double Precision Shared Memory One Sided Jacobi SVD"
	DFASSS	- "Double Precision Full Matrix Add, Shared x s = s"
	DFLUVS	- "Double Precisioin Shared LU factorization"
	DFUTVS	- "Double Precision Shared Upper Triang. Vector Solve"
	DFLTVS	- "Double Precision Shared Lower Triang. Vector Solve"
	DFMLSS	- "Double Precision Full Matrix Mult. (loc.)x(sh.)=(sh.)"
	DFMSLS	- "Double precision full matrix multiply (sh.)x(loc.)=(sh.)"
	DDFSSS	- "double precision diag. matrix mult. (sh.)x(sh.)=(sh.)"
	DFTSSS	- "double prec. full transpose mult. (sh.)x(sh.)=(sh.)"
	DPMALS	- "double prec. print matrix allocation summary"
	DFMSSS	- "double precision full matrix mult. (sh.)x(sh.)=(sh.)"
	DFMCSS	- "double precision full matrix copy (sh.) into (sh.)"
	DMINIS	- "double precision initialize matrix in shared memory"
	DSHALM	- "double precision allocate matrix"
*	DFATSA	- "doubel precision form A(t) x S x A where S is Diagonal"
	DSAINI	- "double precision shared memory initialize package"
	DMVSHR	- "double precision move matrix to / from shared"

DMVTSH - "double precision move transposed to/from shared"

DFMSCA - "double precision full matrix scale"

DSHINP - "double precision shared inner product"

SHORCH - "double precision shared orthogonality check"

SHUTCH - "double precision shared upper triangular check"

DMCPSS - "double precision copy submatrix (sh.) into (sh.)"

PROINI - " initialize package for np processors"

DBGINI - "initialize package for number of processors to debug"

SWLAND - "stop and wait for logical and of all processors"

GNPROC - "query number of processors initialized"

STITST - "set title for timing statistics"

PTITST - "print timing statistics"

OTHER APPLICATIONS UTILITIES

MWSURD - "matrix weighted sum, real double precisioin"

MATRRD - "matrix transpose, real double precision"

MMULRD - "matrix multiply, real double precision"

MAOURD - "matrix output, real double precision"

MACORD - "matrix copy, real double precision"

MAZERD - "matrix zero, real double precision"

LOWER LEVEL MODULES

* GJROSQ - "generate jacobi sequence"

* COSJSV - "concurrent one sided jacobi svd, processor level"

CGQRSQ - "columnwise givens qr sequence generate"

GGQRSQ - "greedy givens qr sequence generate"

* CGQRDP - "concurrent givens qr decomposition, processor level"

* AHLDSH - "concurrent householder qr decomposition, processor level"

* CMGSDP - "concurrent modified gram-schmidt, processor level"

* CONCURRENT MODE

APPLICATIONS PROGRAMMING

LOCATION OF FILES:

SOURCE

/u/user/mefccak/utlshr/*.*

OBJECT LIBRARY

/u/user/mefccak/utlshr/obj/utllib.o

INCLUDE FILES

/u/user/mefccak/dyninc/*.inc

PROGRAMMING USAGE:

To employ the routines in this package, an applications program must carry out the following three steps:

(1) Add a line the the main program including the necessary shared memory defintions.

```
INCLUDE '/u/user/mefccya/dyninc/utlshr.inc'
```

(2) Add a line in the main program to initialize the shared utilities library.

```
CALL USHINI(no_processors)
```

(3) Link the main program with the UTLSHR object module library.

```
shrlib=/u/user/mefccak/utlshr/obj/utllib.o
cf77 -o executable.o application.cf  "$shrlib"
```

ROUTINE

DSHMGS - "DOUBLE PRECISION SHARED MODIFIED GRAM-SCHMIDT"

DESCRIPTION

This subroutine calculates an orthogonal decomposition of a general, rectangular matrix via the modified Gram-Schmidt procedure.

ACCESS

CALL DSHMGS(MA,MB,ICOLM,NTICK,IERR)

ARGUMENTS

MA.... integer,input, matrix no. assigned by dshalm

MB.... integer*4,input, matrix no. assigned by dshalm

ICOLM. integer*4,input, column mode key (1=yes, 0=no)

NTICK. integer*4,output, no. of clock interrupts

IERR.. integer*4, output, error code (1=error, 0=ok)

INCLUDES

SHRMGS Shared variables definition for mgs procedure

SHRPRO Shared variables definition for processors

SHRSYN Shared variables definition for synchronization

SHRMAL Shared variables definition for matrix allocation

EXTERNALS

CMGSDP Concurrent modified Gram-Schmidt processor

CFRTIC Calculate real time ticks (system call)

ROUTINE

DSHCQR - "DOUBLE PRECISION SHARED COLUMNWISE QR DECOMPOSITION"

DESCRIPTION

This routine calculates the QR decomposition of a general, rectangular matrix using Givens transformations.

ACCESS

CALL DSHCQR(MA,MQ1,MQ2,MU,ICOLM,NTICK,IERR)

ARGUMENTS

MA.... integer*4,input, matrix no. assigned by dshalm

MQ1... integer*4,output, matrix no. of ob basis for range

MQ2... integer*4,output, matrix no. of ob basis for nullsp

ICOLM. integer*4,input, column mode key (1=yes, 0=no)

NTICK. integer*4,output, no. of clock interrupts

IERR.. integer*4, output, error code (1=error, 0=ok)

INCLUDES

SHRGQR Shared variables definition for gqr procedure

SHRPRO Shared variables definition for processors

SHRSYN Shared variables definition for synchronization

SHRMAL Shared variables definition for matrix allocation

SHRDBG Shared variables definition for debugging

SHRBUF Shared variables definition for data

EXTERNALS

CGQRSQ Calculate givens qr sequence

CGQRDP Concurrently apply givens qr transform

CFRTIC Calculate real time ticks (system call)

ROUTINE
DFASSS - "DOUBLE PRECISION FULL MATRIX ADD IN SHARED"

DESCRIPTION
This routine adds two matrices in shared memory, ie, (shared) X (shared) ======¿ (shared)

ACCESS
CALL DFASSS(M1,M2,ICONC,IERR)

ARGUMENTS

M1.... integer*4,input, matrix no. of first matrix

M2.... integer*4,in/out, matrix no. of second matrix

ICONC. integer*4,input, concurrency switch (1=yes)

IERR.. integer*4, output, error code (1=error, 0=ok)

INCLUDES

SHRMAL Shared variables definition for matrix allocation

SHRBUF Shared variables definition for data

EXTERNALS

NONE

ROUTINE
DFLUVS - "DOUBLE PRECISION SHARED LU FACTORIZATION"

DESCRIPTION
This subroutine calculates the lu factorization of a general, densely filled matrix.

ACCESS
CALL DFLUVS(MA,MB,ICONC,IFACT,NTICK,IERR)

ARGUMENTS

MA.... integer*4,in/out, matrix no. assigned by dshalm

MB.... integer*4,in/out, vector no. assigned by dshalm

ICONC. integer*4,input , concurrency switch (1=yes,0=no)

IFACT. integer*4,input , 1=factor matrix, 0=just f-sub,b-sub

NTICK. integer*4,output, no. of clock interrupts

IERR.. integer*4, output, error code (1=error, 0=ok)

INCLUDES

SHRBUF Shared variables definition for data buffer

SHRMAL Shared variables definition for matrix allocation

EXTERNALS

NONE

ROUTINE

DFUTVS - "DOUBLE PRECISION UPPER TRIANGULAR - VECTOR SOLVE"

DESCRIPTION

This subroutine solves a linear system of equations when the coefficient is a upper triangular matrix. In other words, a back substitution is carried out.

ACCESS

CALL DFUTVS(MU,MR,ICONC,NTICK,IERR)

ARGUMENTS

MU.... integer,input, u.t. matrix no. assigned by dshalm

MR.... integer*4,input, rhs vector assigned by dshalm

ICONC. integer*4,input, concurrency switch (1=yes)

NTICK. integer*4,output, no. of clock interrupts

IERR.. integer*4, output, error code (1=error, 0=ok)

INCLUDES

SHRBUF Shared variables definition for data buffer

SHRMAL Shared variables definition for matrix allocation

EXTERNALS

NONE ...

ROUTINE
DFLTVS - "DOUBLE PRECISION LOWER TRIANGULAR - VECTOR SOLVE"

DESCRIPTION
This subroutine solves a linear system of equations when the coefficient matrix is a lower triangular matrix. In other words, a forward substitution is carried out.

ACCESS
CALL DFLTVS(ML,MR,ICONC,NTICK,IERR)

ARGUMENTS

ML.... integer*4,input, l.t. matrix no. assigned by dshalm

MR.... integer*4,input, rhs vector assigned by dshalm

ICONC. integer*4,input, concurrency switch (1=yes)

NTICK. integer*4,output, no. of clock interrupts

IERR.. integer*4, output, error code (1=error, 0=ok)

INCLUDES

SHRBUF Shared variables definition for data buffer

SHRMAL Shared variables definition for matrix allocation

EXTERNALS

NONE ...

ROUTINE
DFMLSS - "DOUBLE PRECISION FULL MATRIX MULTIPLY (LSS)"

DESCRIPTION
This subroutine multiplies a (local) matrix times a (shared) matrix and stores the result in (shared)

ACCESS
CALL DFMLSS(A,M,N,M2,M3,ICONC,IERR)

ARGUMENTS

A..... real*8,input, local mxn matrix

M..... integer*4,input, no. rows in A

N..... integer*4,input, no. cols in A

M2.... integer*4,input, matrix no. of 2nd matrix

M3.... integer*4,input, matrix no. of 3rd matrix

ICONC. integer*4,input, concurrency switch (1=yes)

IERR.. integer*4, output, error code (1=error, 0=ok)

INCLUDES

SHRBUF Shared variables definition for data buffer

SHRMAL Shared variables definition for matrix allocation

EXTERNALS

NONE ...

ROUTINE

DDFSSS - "DOUBLE PRECISION DIAGONAL X FULL MATRIX MULTIPLY (SSS)"

DESCRIPTION

This subroutine multiplies a (shared) diagonal matrix times a (shared) full matrix and stores the result in (shared)

ACCESS

CALL DDFSSS(M1,M2,IERR)

ARGUMENTS

M1.... integer*4,input, matrix no. of diagonal matrix

M2.... integer*4,input, matrix no. of full matrix

IERR.. integer*4, output, error code (1=error, 0=ok)

INCLUDES

SHRBUF Shared variables definition for data buffer

SHRMAL Shared variables definition for matrix allocation

EXTERNALS

NONE ...

ROUTINE

DFTSSS - "DOUBLE PRECISION FULL TRANSPOS MATRIX MULTIPLY (SSS)"

DESCRIPTION

This subroutine multiplies a (shared) transposed matrix times a (shared) full matrix and stores the result in (shared)

ACCESS

CALL DFTSSS(M1,M2,M3,ICONC,IERR)

ARGUMENTS

M1.... integer*4,input, matrix no. of transposed matrix

M2.... integer*4,input, matrix no. of full matrix

M3.... integer*4,input, matrix no. of product matrix

IERR.. integer*4, output, error code (1=error, 0=ok)

INCLUDES

SHRBUF Shared variables definition for data buffer

SHRMAL Shared variables definition for matrix allocation

EXTERNALS

NONE ...

ROUTINE
DPMALS - "DOUBLE PRECISION PRINT MATRIX ALLOCATION"

DESCRIPTION
This subroutine prints a summary of all allocated matrices.

ACCESS
CALL DPMALS(LOUT,IERR)

ARGUMENTS

LOUT.. integer*4,input, logical unit no. for output

IERR.. integer*4, output, error code (1=error, 0=ok)

INCLUDES

SHRBUF Shared variables definition for data buffer

SHRMAL Shared variables definition for matrix allocation

EXTERNALS

NONE ...

ROUTINE
DFMSSS - "DOUBLE PRECISION FULL MATRIX MULTIPLY (SSS)"

DESCRIPTION
This subroutine multiplies a full (shared) matrix times a full (shared) matrix and stores the result in (shared).

ACCESS
CALL DFMSSS(M1,M2,M3,ICONC,IERR)

ARGUMENTS

M1.... integer*4,input, 1st matrix no.

M2.... integer*4,input, 2nd matrix no.

M3.... integer*4,input, 3rd matrix no.

ICONC. integer*4,input, concurrency switch, 1=yes

IERR.. integer*4, output, error code (1=error, 0=ok)

INCLUDES

SHRBUF Shared variables definition for data buffer

SHRMAL Shared variables definition for matrix allocation

EXTERNALS

NONE ...

ROUTINE
DFMCSS - "DOUBLE PRECISION FULL MATRIX COPY SHARED INTO SHARED"

DESCRIPTION
This subroutine copies a full (shared) matrix into a full (shared) matrix.

ACCESS
CALL DFMCSS(MA,MB,IERR)

ARGUMENTS

MA.... integer*4,input, 1st matrix no.

MB.... integer*4,input, 2nd matrix no.

IERR.. integer*4, output, error code (1=error, 0=ok)

INCLUDES

SHRBUF Shared variables definition for data buffer

SHRMAL Shared variables definition for matrix allocation

EXTERNALS

NONE ...

ROUTINE
DATSAP - "DOUBLE PRECISION FULL MATRIX A^TSA "

DESCRIPTION
This subroutine forms the matrix triple product A^TSA where A is a full matrix and S is a diagonal matrix.

ACCESS
CALL DATSAP(IA,IS,IASA,M,N,IP,NP)

ARGUMENTS

IA.... integer*4,input, pointer to matrix A

IS.... integer*4,input, pointer to matrix S

IASA.. integer*4,input, pointer to triple product

M..... integer*4,input, no. of rows in A

N..... integer*4,input, no. of cols in A

IP.... integer*4,input, processor no.

NP.... integer*4,input, no. of processors

INCLUDES

SHRBUF Shared variables definition for data buffer

SHRMAL Shared variables definition for matrix allocation

EXTERNALS

SWLAND SYNCHRONIZATION ROUTINE

ROUTINE

DMINIS - "DOUBLE PRECISION MATIX INITIALIZE"

DESCRIPTION

This subroutine initializes a matrix in shared to a desired value.

ACCESS

CALL DMINIS(NM,DINI,ICONC,IERR)

ARGUMENTS

NM.... integer*4,input, matrix no. to initialize

DINI.. real*8,input, value to initialize matrix

ICONC. integer*4,input, concurrency switch, 1=yex

IERR.. integer*4,output, error code 1=err,0=ok

INCLUDES

SHRBUF Shared variables definition for data buffer

SHRMAL Shared variables definition for matrix allocation

EXTERNALS

NONE

ROUTINE

DSHALM - "DOUBLE PRECISION SHARED MATRIX ALLOCATE"

DESCRIPTION

This subroutine allocates a matrix for use with most of the other subroutines in this package.

ACCESS

CALL DSHALM(IMTYPE,NROW,NCOL,MANAME,LCOL,NP,IP,IERR)

ARGUMENTS

IMTYPE integer*4,input, matrix type (1=full)

NROW.. integer*4,input, no. of rows in matrix

NCOL.. integer*4,input, no. of cols in matrix

MANAME charac*40,input, matrix name

LCOL.. NOT USED CURRENTLY

NP.... integer*4,output, matrix no. assigned

IP.... integer*4,output, pointer to matrix in shared

IERR.. integer*4,output, error code 1=err,0=ok

INCLUDES

SHRBUF Shared variables definition for data buffer

SHRMAL Shared variables definition for matrix allocation

EXTERNALS

NONE

ROUTINE

DFATSA - "DOUBLE PRECISION FORM $A^T SA$ "

DESCRIPTION

This subroutine forms the matrix triple product $A^T SA$ where S is a diagonal matrix and A is a full matrix.

ACCESS

CALL DFATSA(MA,MS,MB,ICONC,NTICK,IERR)

ARGUMENTS

MA.... integer*4,input, matrix no. of A

MS.... integer*4,input, matrix no. of S

MB.... integer*4,input, matrix no. of B

ICONC. integer*4,input, concurrency switch

NTICK. integer*4,output, no. of clock interuppts

IERR.. integer*4,output, error code 1=err,0=ok

INCLUDES

SHRBUF Shared variables definition for data buffer

SHRMAL Shared variables definition for matrix allocation

SHRPRO Shared variables definition for processors

SHRSYN Shared variables definition for synchronization

SHRINT Shared variables definition for integer buffer

EXTERNALS

DATSAP Form matrix product on each processor

ROUTINE
DSAINI - "DOUBLE PRECISION SHARED INITIALIZE"

DESCRIPTION
This subroutine initializes the matrix allocation package for double precision shared utilities.

ACCESS
CALL DSAINI

ARGUMENTS

NONE

INCLUDES

SHRBUF Shared variables definition for data buffer

SHRMAL Shared variables definition for matrix allocation

EXTERNALS

NONE

ROUTINE
PROINI - "PROCESSORS INITIALIZE"

DESCRIPTION
This subroutine initializes the processor data for the double precision package.

ACCESS
CALL PROINT(NP)

ARGUMENTS

NP.... integer*4,input,no. of processors to be used

INCLUDES

SHRPRO Shared variables definition for processors

SHRSYN Shared variables definition for synchronization

EXTERNALS

NONE

ROUTINE

DMVSHR - "DOUBLE PRECISION MOVE MATRIX TO/FROM SHARED"

DESCRIPTION

This subroutine moves a matrix to/from local to shared memory.

ACCESS

CALL DMVSHR(DM,M,N,IPTR,IRW,IERR)

ARGUMENTS

DM.... real*8, input, double precision local matrix

M..... integer*4,input,no. of rows in dm

N..... integer*4,input,no. of cols in dm

IPTR.. integer*4,input,pointer to matrix in shared

IRW... integer*4,input,read/write key (1=copy to,2=copy from)

IERR.. integer*4,output,error code 1=err,0=ok

INCLUDES

SHRBUF Shared variables definition for data

EXTERNALS

NONE

ROUTINE

DMVTSH - "DOUBLE PRECISION MOVE TRANSPOSED MATRIX TO/FROM SHARED"

DESCRIPTION

This subroutine moves a transposed matrix to/from local to shared memory.

ACCESS

CALL DMVTSH(DM,M,N,IPTR,IRW,IERR)

ARGUMENTS

DM.... real*8, input, double precision local matrix

M..... integer*4,input,no. of rows in dm

N..... integer*4,input,no. of cols in dm

IPTR.. integer*4,input,pointer to matrix in shared

IRW... integer*4,input,read/write key (1=copy to,2=copy from)

IERR.. integer*4,output,error code 1=err,0=ok

INCLUDES

SHRBUF Shared variables definition for data

EXTERNALS

NONE

ROUTINE
DFMSCA - "DOUBLE PRECISION FULL MATRIX SCALE"

DESCRIPTION
This subroutine scales a full matrix in shared memory.

ACCESS
CALL DFMSCA(MA,SCALE,ICONC,IERR)

ARGUMENTS

MA.... integer*4,input,matrix no. to scale

SCALE. real*8,input,scale factor

ICONC. integer*4,input,concurrency switch 1=yes,0=no

IERR.. integer*4,output,error code 1=err,0=ok

INCLUDES

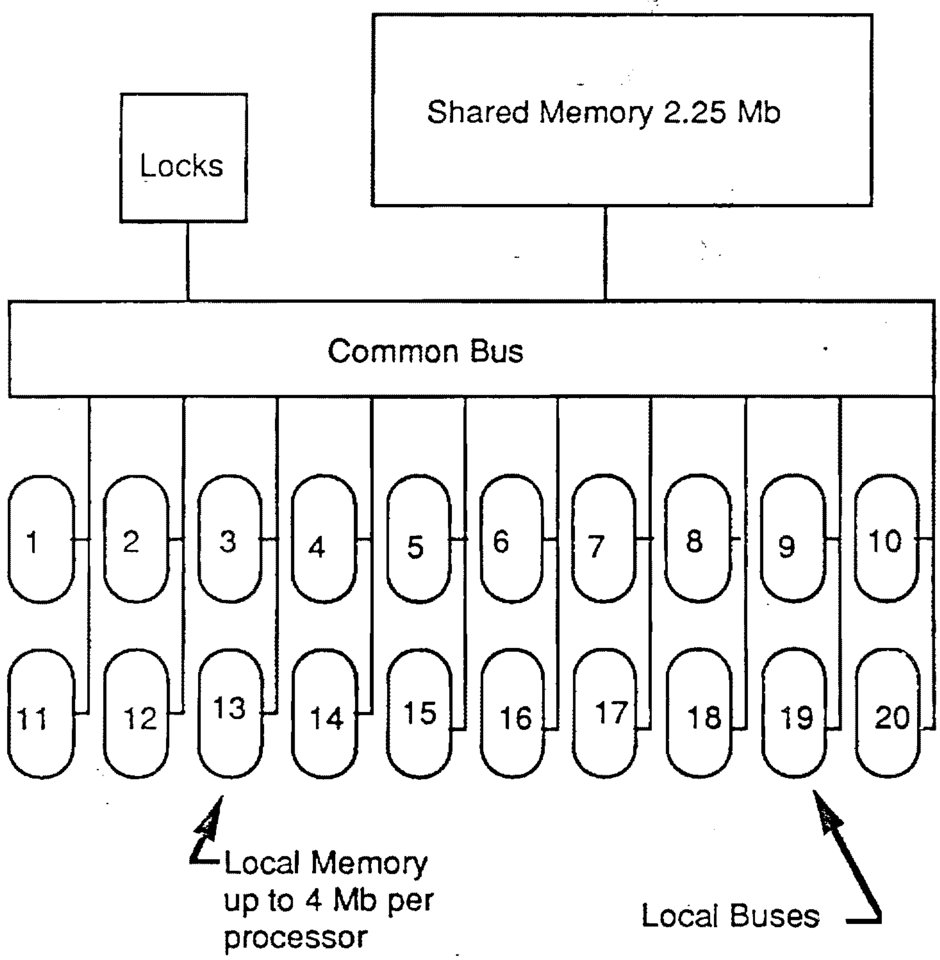
SHRBUF Shared variables definition for data

SHRMAL Shared variables definition for matrix allocation

EXTERNALS

NONE

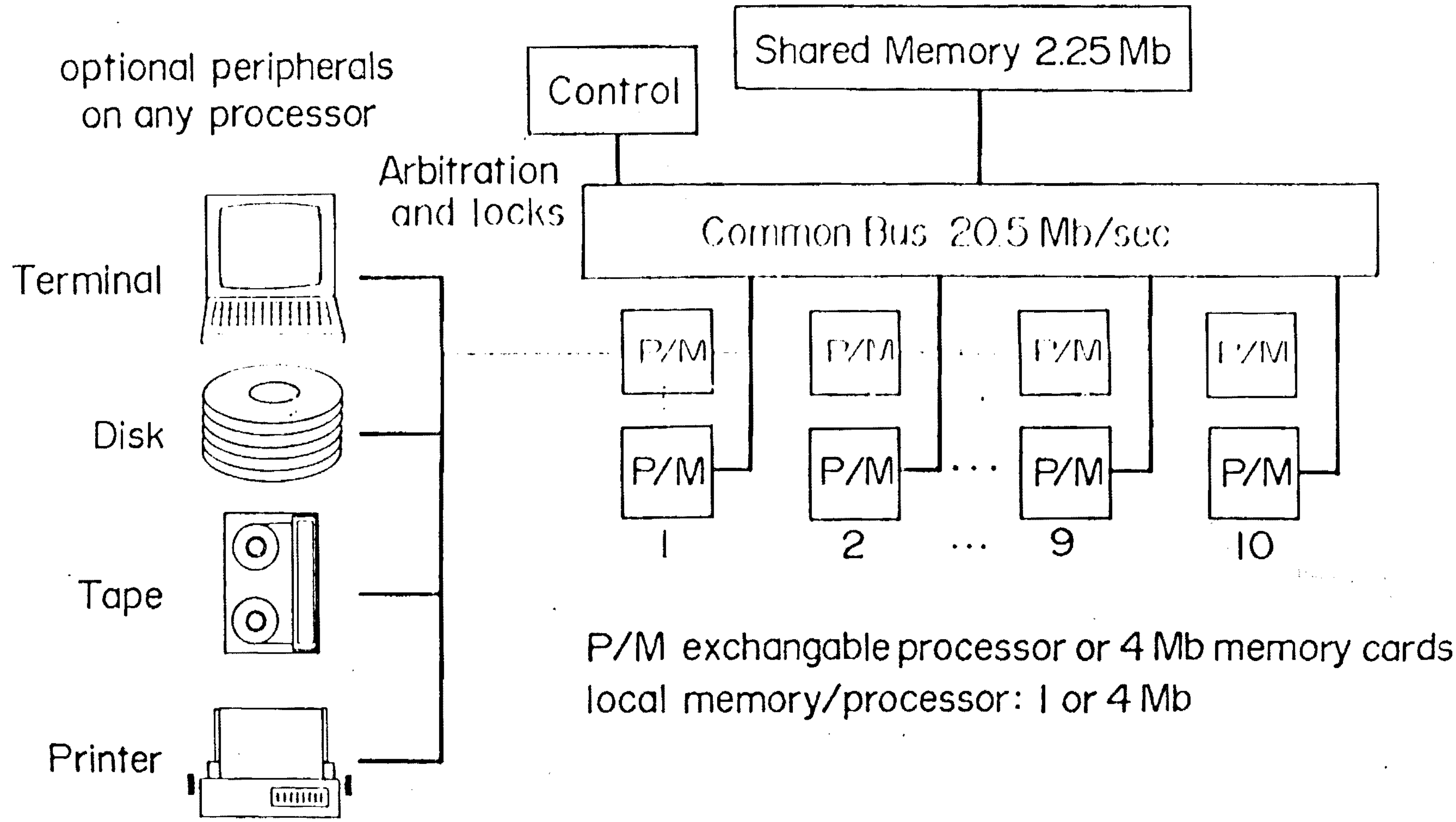
PRESENTATION BY
CHRIS AYERS



Flex 32 Architecture Schematic
for 20 Processors

Figure (4.1)

FLEX/32 SHARED MEMORY ARCHITECTURE



QR DECOMPOSTION

$[A] = [Q][R]$ Decompose [A] into two matrices [Q] and [R]

[A]-----> M x N
[Q]-----> M x M (orthogonal)
[R]-----> M x N (upper triangular)

METHOD: HOUSEHOLDER TRANSFORM

$Q_{(K)} = (I - (2vv^T)/(v^Tv))$

Where v is found from the following algorithm

$m: =\max\{/x_k/,\ldots\ldots\ldots,/x_j/\}$
$\alpha: = 0$
For i=k to j
$v_i := x_i/m$
$\alpha := \alpha + v_i^2$
$\alpha := \sqrt{\alpha}$
$\beta := 1/(\alpha\,(\alpha + /v_k/))$
$v^T := v_k + \text{sign}(v_k)\alpha$

Since the algorithm was implemented concurrently $Q_{(k)}$ was never explicitly formed. It was implemented in vector from.

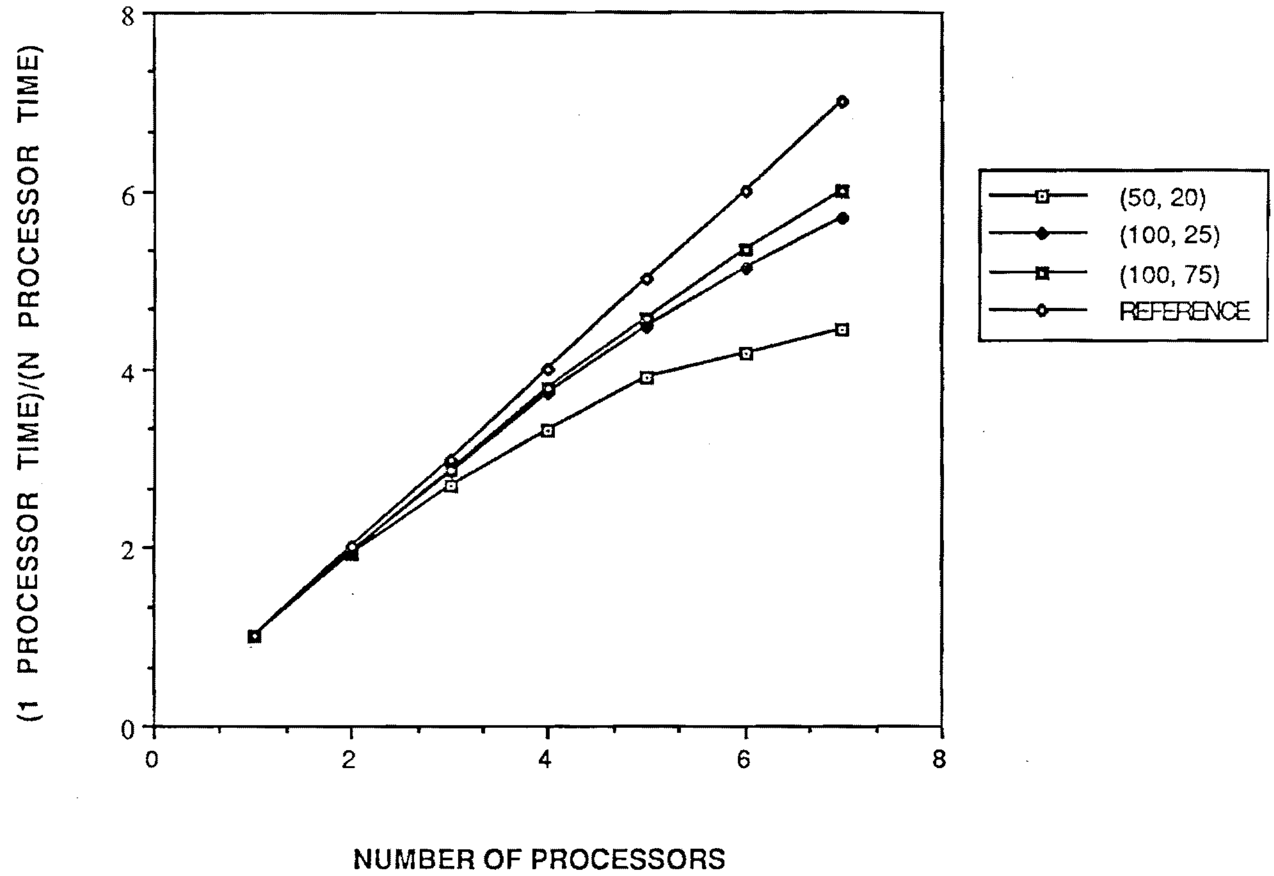
$R = (Q_n^T Q_{n-1}^T \ldots\ldots\ldots\ldots Q_2^T Q_1^T)\,[A]$
$Q = [I]\,(Q_1 Q_2 Q_3 \ldots\ldots\ldots\ldots Q_{n-1} Q_n)$

ADVANTAGES:

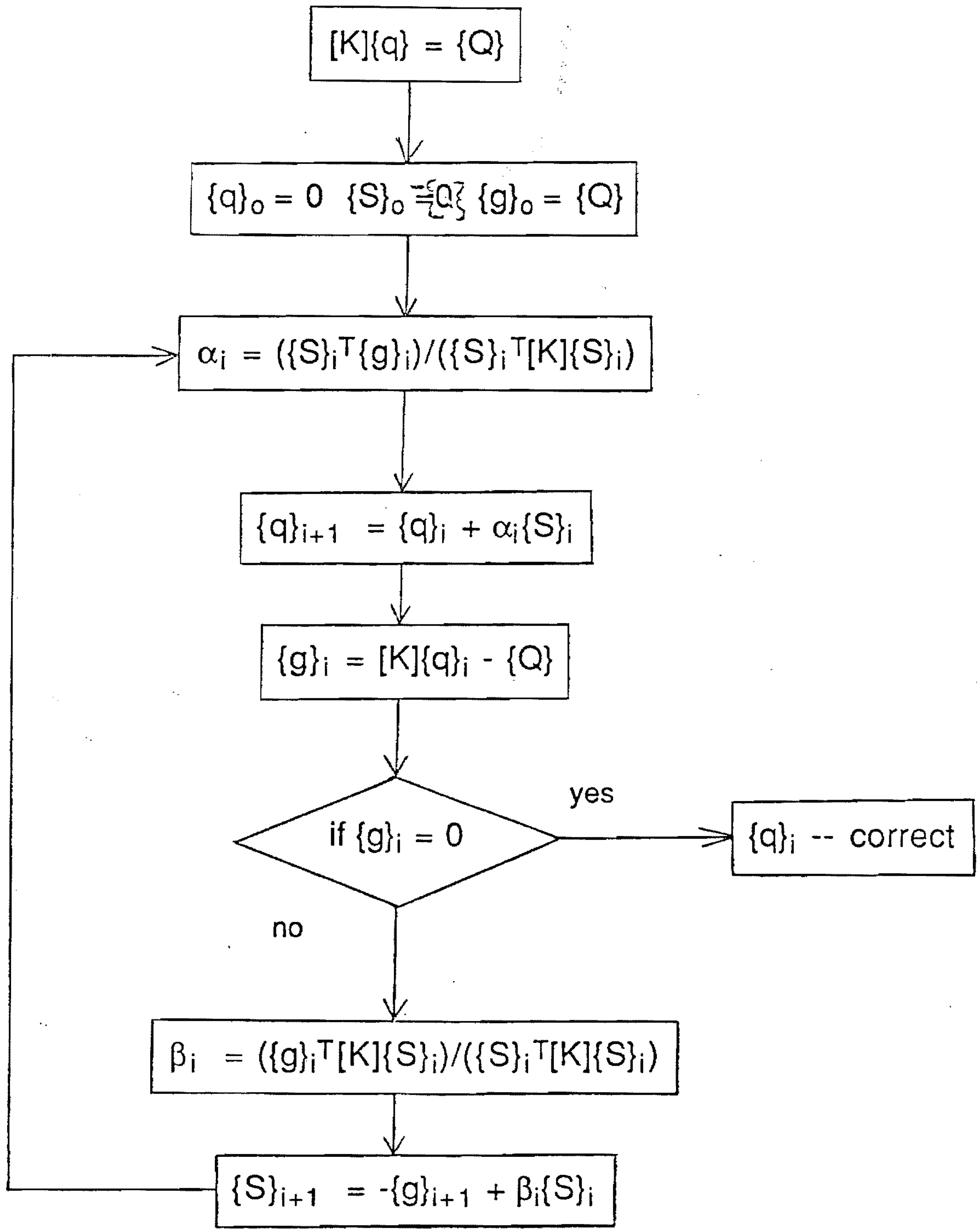
$[A]\,\{q\} = \{F\}$
$[Q]\,[R]\,\{q\} = \{F\}$
$[R]\,\{q\} = [Q^T]\,\{F\}$ Since Q is orthogonal $Q^{-1} = Q^T$

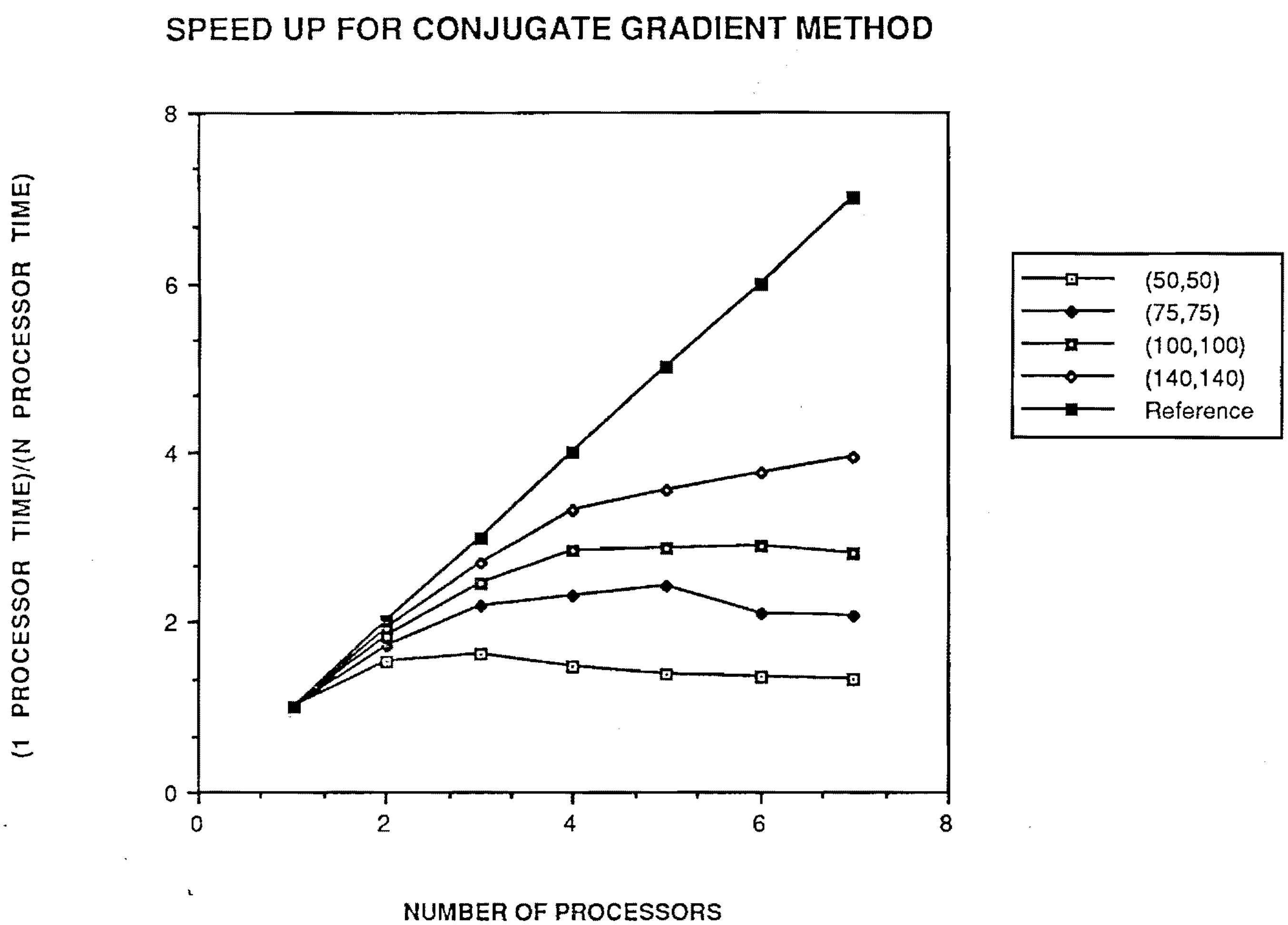
The vector {q} can now be found from forward substitution without having to take the inverse of the [A] matrix.

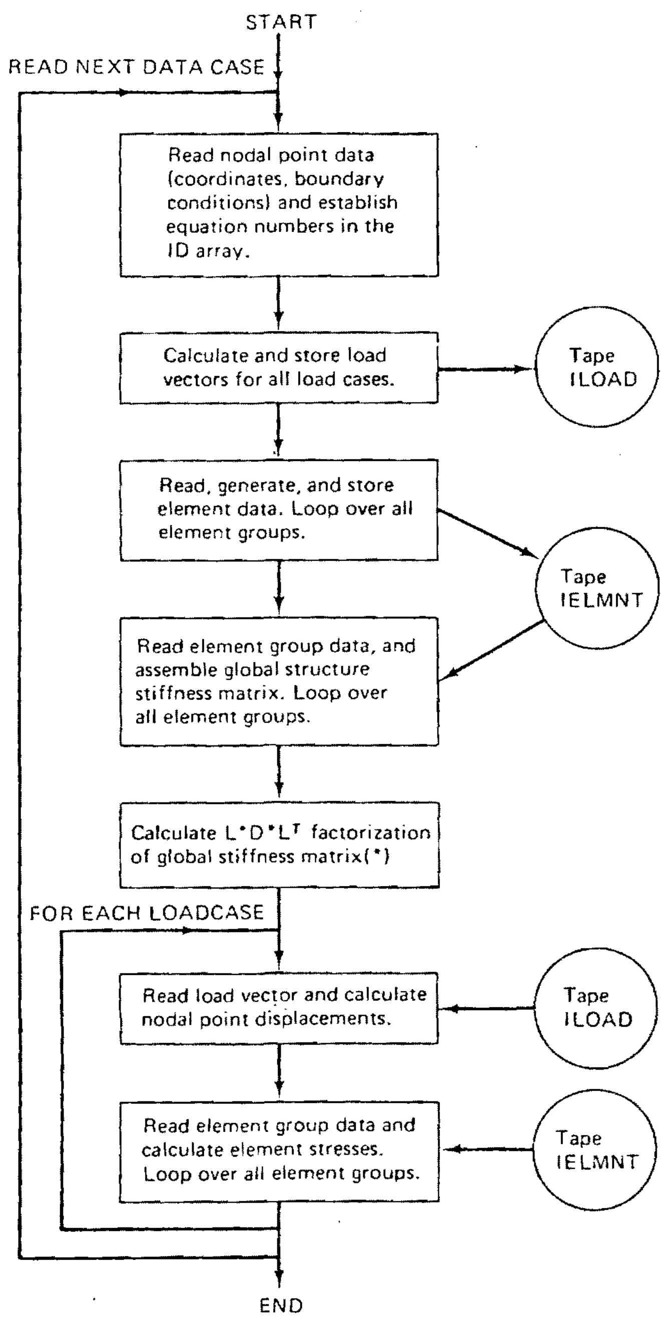
SPEED UP OF QR DECOMPOSITION OF MATRIX



CONJUGATE GRADIENT METHOD



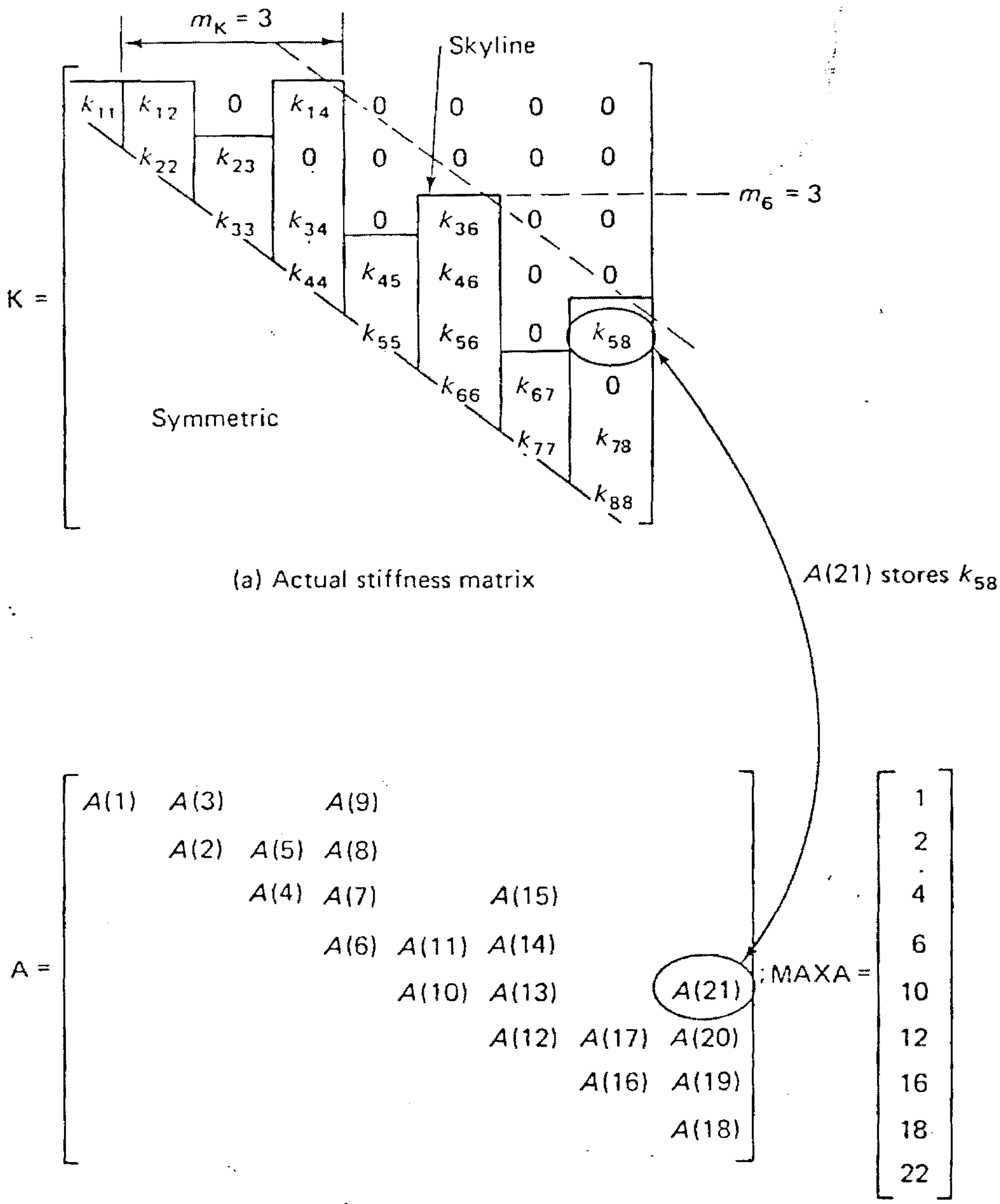

SPEED UP FOR CONJUGATE GRADIENT METHOD
(1 PROCESSOR TIME)/(N PROCESSOR TIME)
NUMBER OF PROCESSORS
0
2
4
6
8
(50,50)
(75,75)
(100,100)
(140,140)
Reference


START
READ NEXT DATA CASE
Read nodal point data (coordinates, boundary conditions) and establish equation numbers in the ID array.
Calculate and store load vectors for all load cases.
Tape ILOAD
Read, generate, and store element data. Loop over all element groups.
Tape IELMNT
Read element group data, and assemble global structure stiffness matrix. Loop over all element groups.
Calculate L*D*L^T factorization of global stiffness matrix(*)
FOR EACH LOADCASE
Read load vector and calculate nodal point displacements.
Tape ILOAD
Read element group data and calculate element stresses. Loop over all element groups.
Tape IELMNT
END

SAPNEW FINITE ELEMENT PROGRAM

```
CALL INPUT
CALL LOADS
CALL ELCAL
        CALL ELEMENT
                CALL SOL21
                        CALL THDFE
                                CALL  INP21*
                                        CALL VECTR2*
                                        CALL CROSS2*
                                CALL SSLAW*
                                CALL  STR821*
                                        CALL DER3DS*
                                CALL FACEPR*
                                        CALL FUNCT*
                                CALL CALBAN*
                                CALL DER3DS*
                                CALL COLHT*
                CALL STRC
                CALL STRC
CALL ADRESS
CALL CLEAR
CALL ASSEM
        CALL ADBAN
CALL COLSOL*
CALL LOADV
CALL COLSOL*
CALL WRITE
CALL STRESS
        CALL ELEMENT
                CALL SOL21
                        CALL THDFE
                                CALL  INP21*
                                        CALL VECTR2*
                                        CALL CROSS2*
                                CALL SSLAW*
                                CALL  STR821*
                                        CALL DER3DS*
                                CALL FACEPR*
                                        CALL FUNCT*
                                CALL CALBAN*
                                CALL DER3DS*
                                CALL COLHT*
                CALL STRC
                CALL STRC
END OF PROGRAM
```

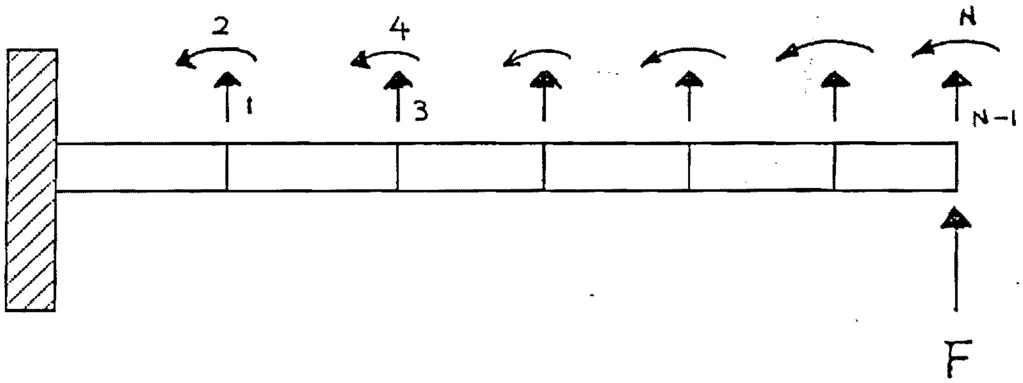
*--->SUBROUTINES THAT WILL BE INVOLVED IN THE CONCURRENT PROCESSING



(a) Actual stiffness matrix

(b) Array A storing elements of K

< Test Model Structure >



(N D.O.F.)

* Element Stiffness Matrix

$$[K]_e = \frac{EI}{L^3}\begin{bmatrix} 12 & 6L & -12 & 6L \\ 6L & 4L^2 & -6L & 2L^2 \\ -12 & -6L & 12 & -6L \\ 6L & 2L^2 & -6L & 4L^2 \end{bmatrix}$$

~~for~~

$$[K]_G = \sum_{i=1} [K]_e^i$$

< The System of Equations >

$$[K]_G \{u\} = \{F\}$$

If $[K]_G$ is a symmetric, positive definite matrix, — for linear elements

$[K]$ may be decomposed into $[L][L^T]$. Then

$$[K]\{u\} = \{F\}$$

$$[L][L]^T\{u\} = \{F\} \quad \text{--------}(a)$$

$$[L]\{y\} = \{F\} \quad \text{--------}(b)$$

$$[L]^T\{u\} = \{y\} \quad \text{--------}(c)$$

(a) Cholesky Decomposition

(b) Forward Substitution
(c) Back Substitution
$\Big\} \rightarrow$ sequential process

* Cholesky Decomposition

A symmetric positive definite matrix $(N \times N)$

K may be decomposed into $L L^T$ using

the following algorithm.

$$L_{ik} = (K_{ik} - \sum_{p=1}^{k-1} L_{ip} L_{kp}) / L_{kk}$$

$$L_{kk} = (K_{kk} - \sum_{p=1}^{k-1} L_{kp}^2)^{1/2} .$$

In kji form,

for $k = 1, N$

$L_{kk} = (K_{kk})^{1/2}$

for $s = k+1, N$

$L_{sk} = K_{sk} / L_{kk}$

for $j = k+1, N$

for $i=j, N$

$K_{ij} = K_{ij} - L_{ik} L_{jk}$

In shorthand

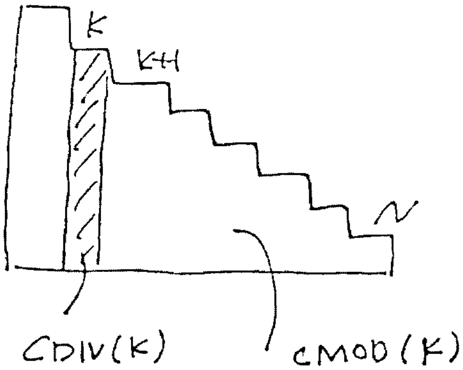
for $k = 1, N$

cdiv(k)

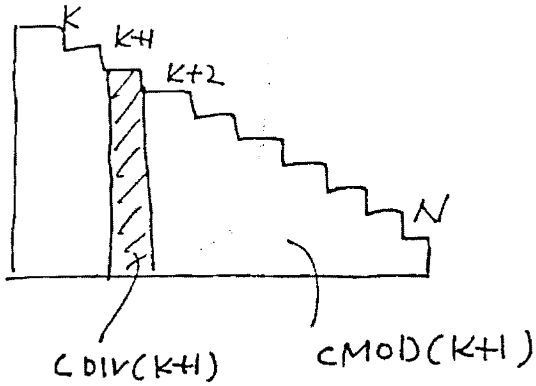
for $j = k+1, N$

cmod(k)

◎ Sequential Operation



$\Rightarrow$



CDIV(1)

$\boxed{\text{for } I = 1, N-1}$

CMOD(I)

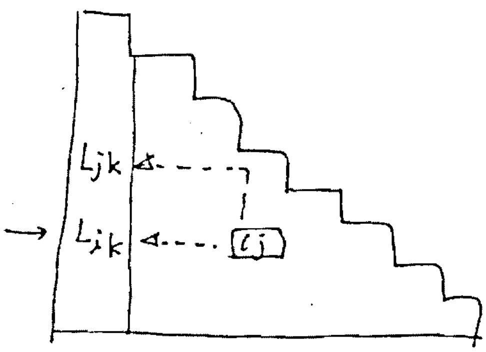
CDIV(I+1)

$\boxed{\text{NEXT } I}$

$$\text{CDIV}(K) \Rightarrow \begin{cases} L_{KK} = (K_{KK})^{1/2} \\ L_{SK} = K_{SK} / L_{KK} \end{cases}$$

CMOD(K)

$$K_{ij} = K_{ij} - L_{ik} * L_{jk}.$$



$\Rightarrow$ ① Column-wise modification.

$$K_{is} = K_{is} - L_{ik} * L_{sk}$$

fixed for the column.

$(i = s, N)$

$\Rightarrow$ ② Row-wise Modification

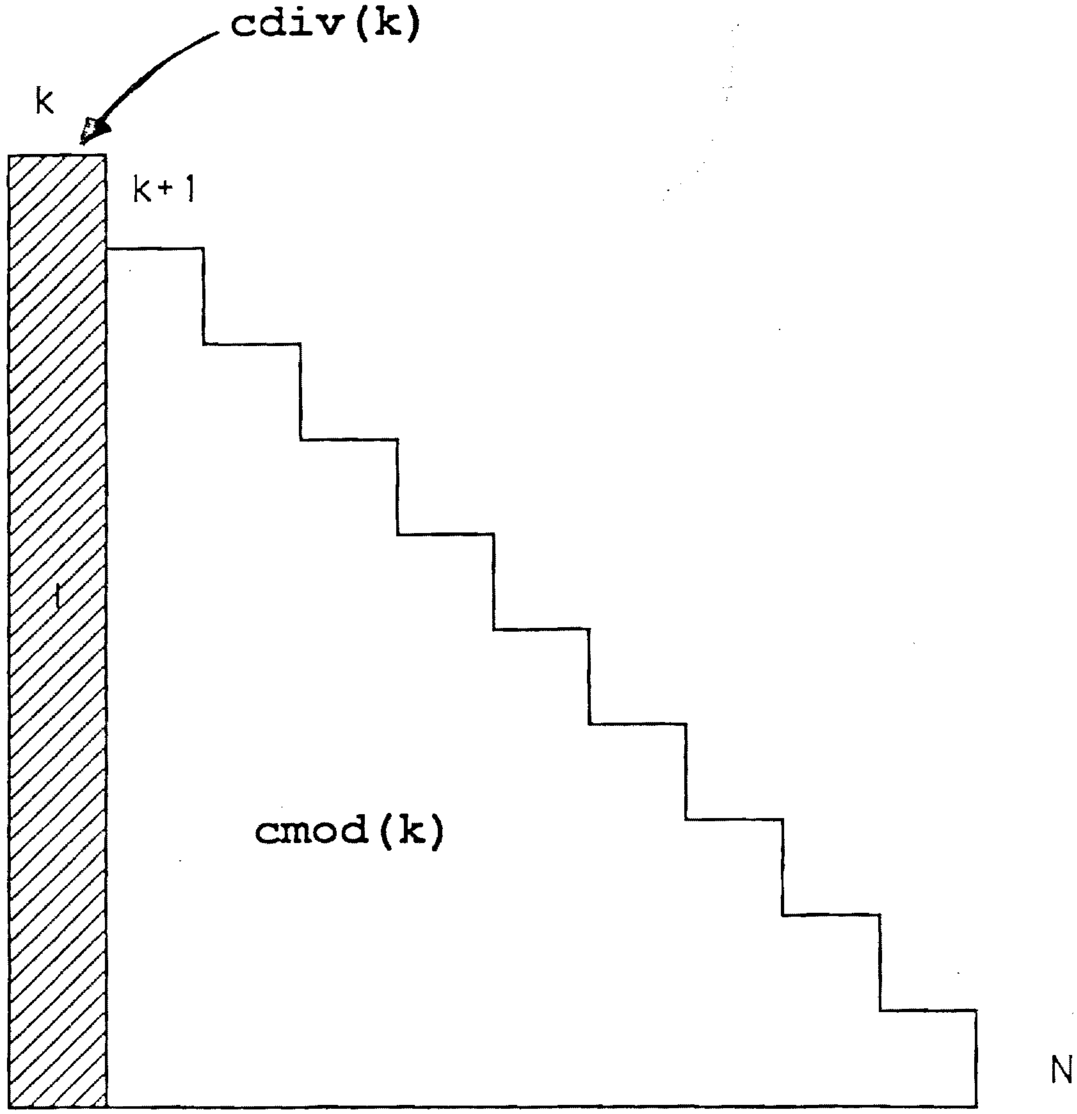
$$K_{sj} = K_{sj} - L_{sk} * L_{jk}$$

fixed for the row

$(j = k+1, s)$

```
* < start >

  cdiv(1)

  for k = 1, N-1

     cmod1(k)

     cdiv(k+1)

     cmod1(k+1)          :wait until completed

     for j = k+2, N

        cmod2(k,j)       :wait until all the
                          processors finish
     next j

  next k

< Forward Substitution >

< Back  Substitution >

< Stop >

< End >
```

* Published in ,"Solution of Structural Analysis Problems on a Parallel Computer",by Storaasli, O., and Ortega, J., (1988)

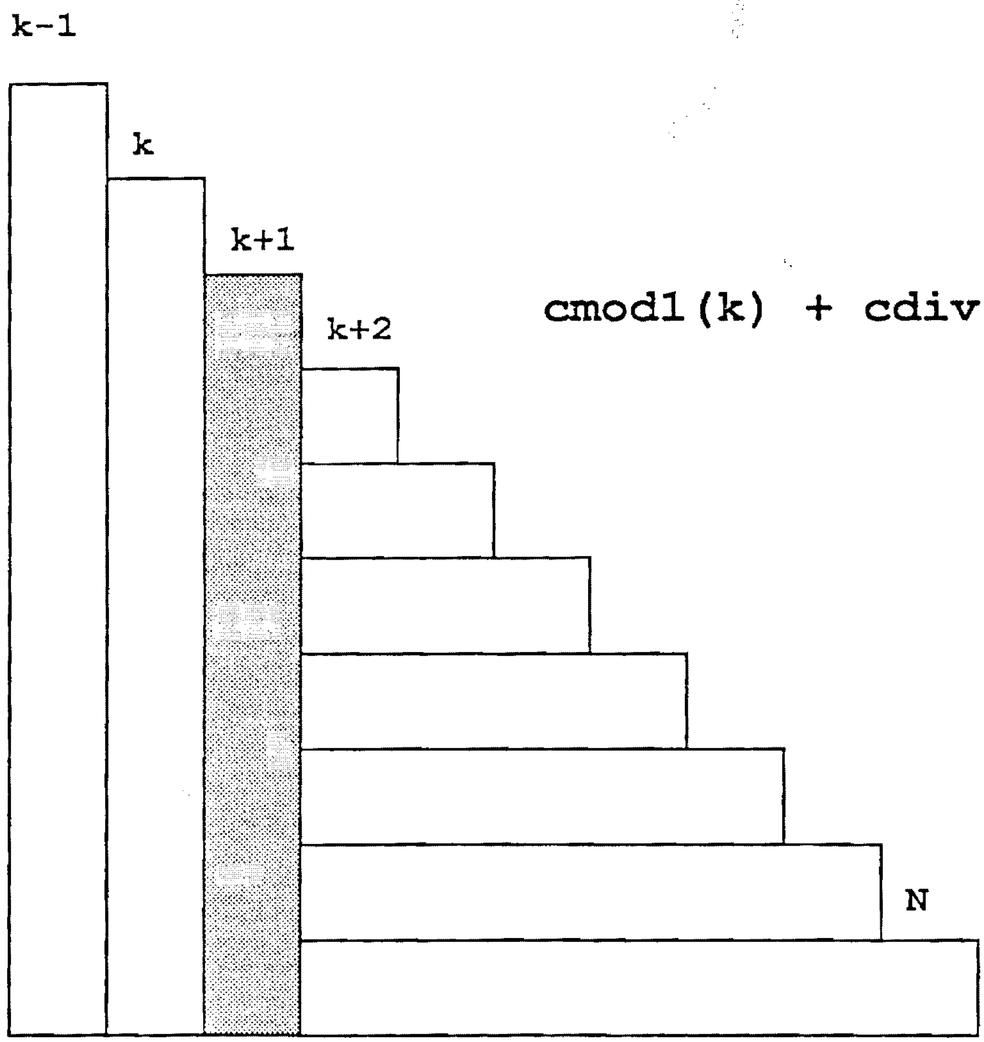


i) cdiv(k)

ii) cmod1(k)

iii) cmod2(k, j)

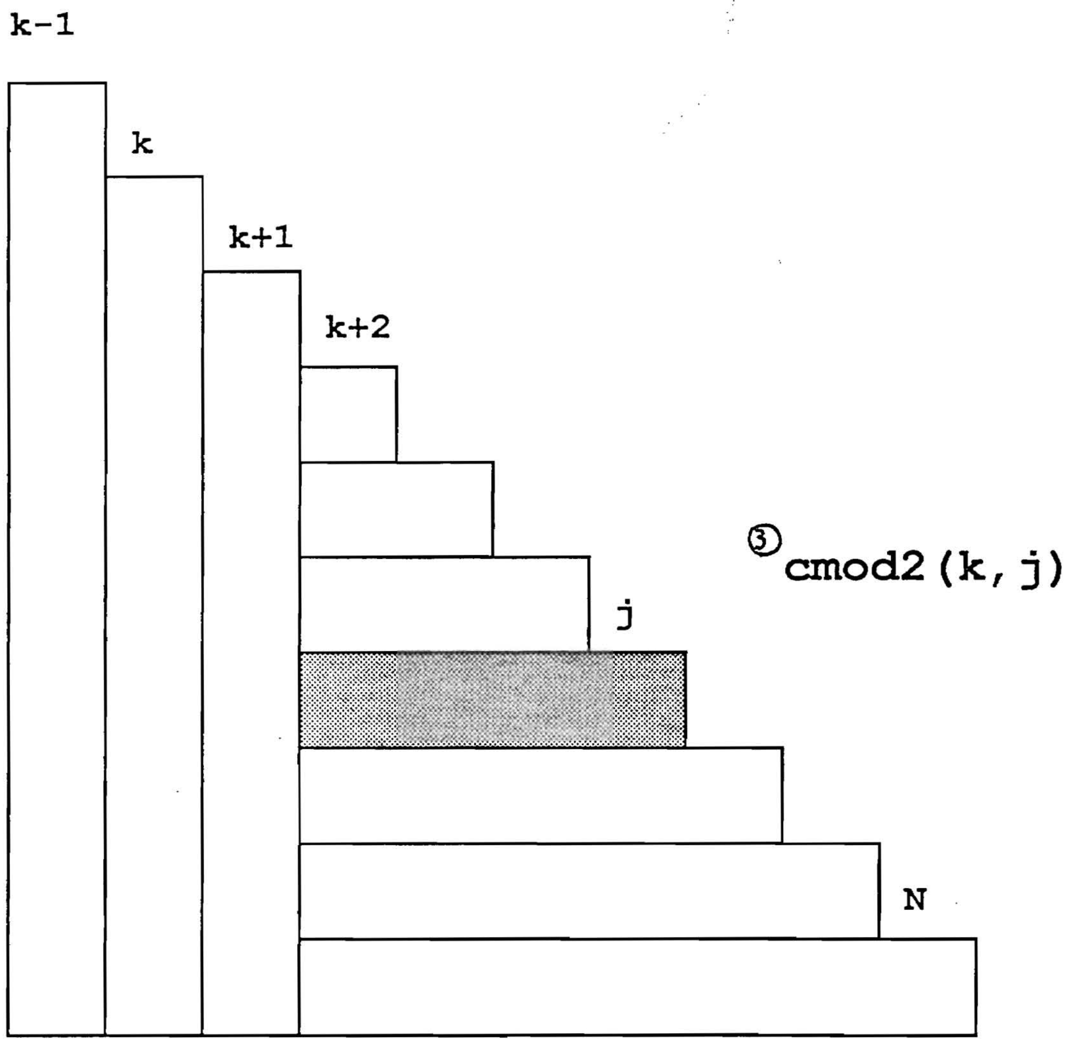
cmod(k)



cmod1(k) + cdiv(k+1)

i) cdiv(k)

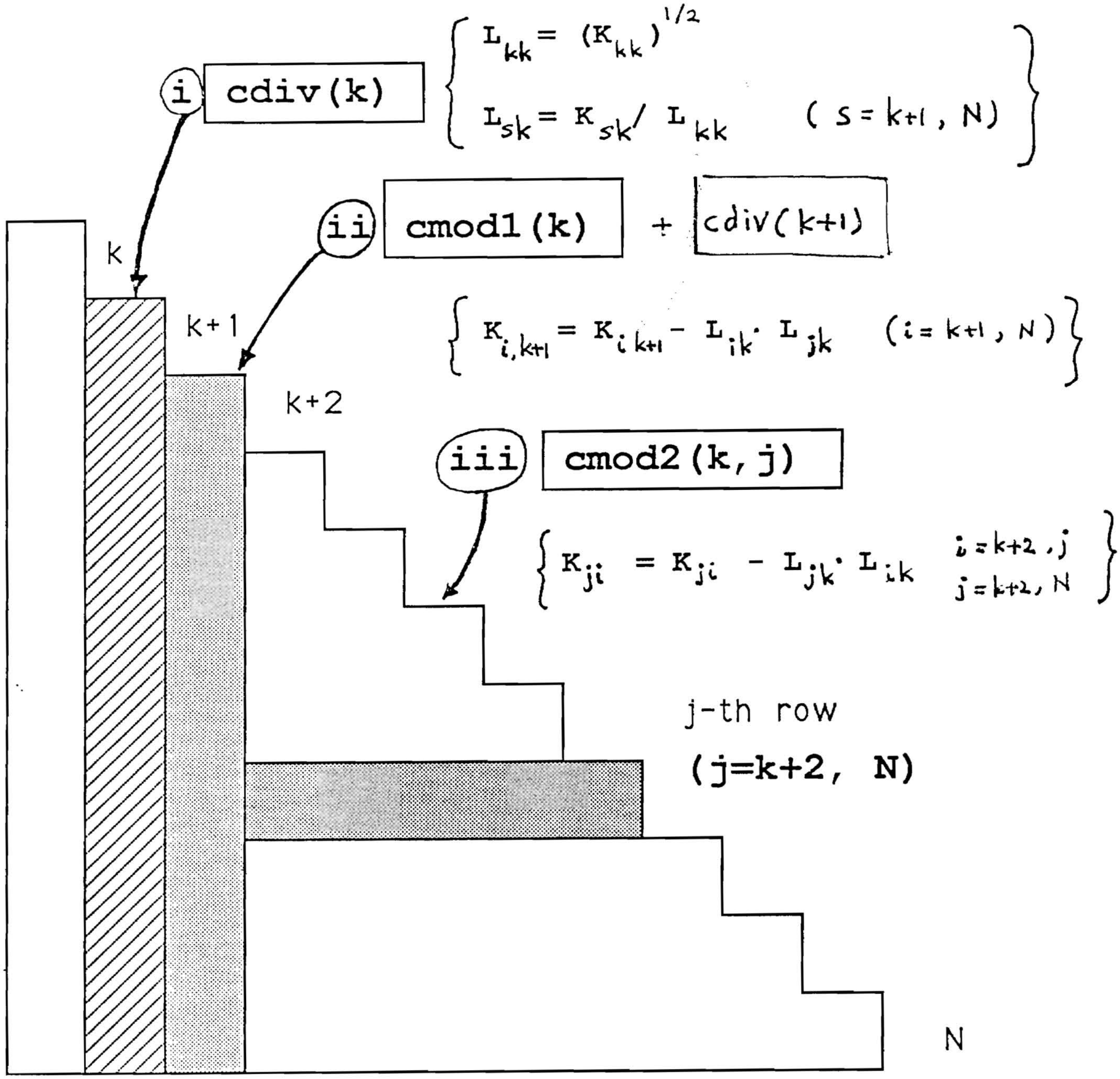
ii) cmod(k)



i) cdiv(k)

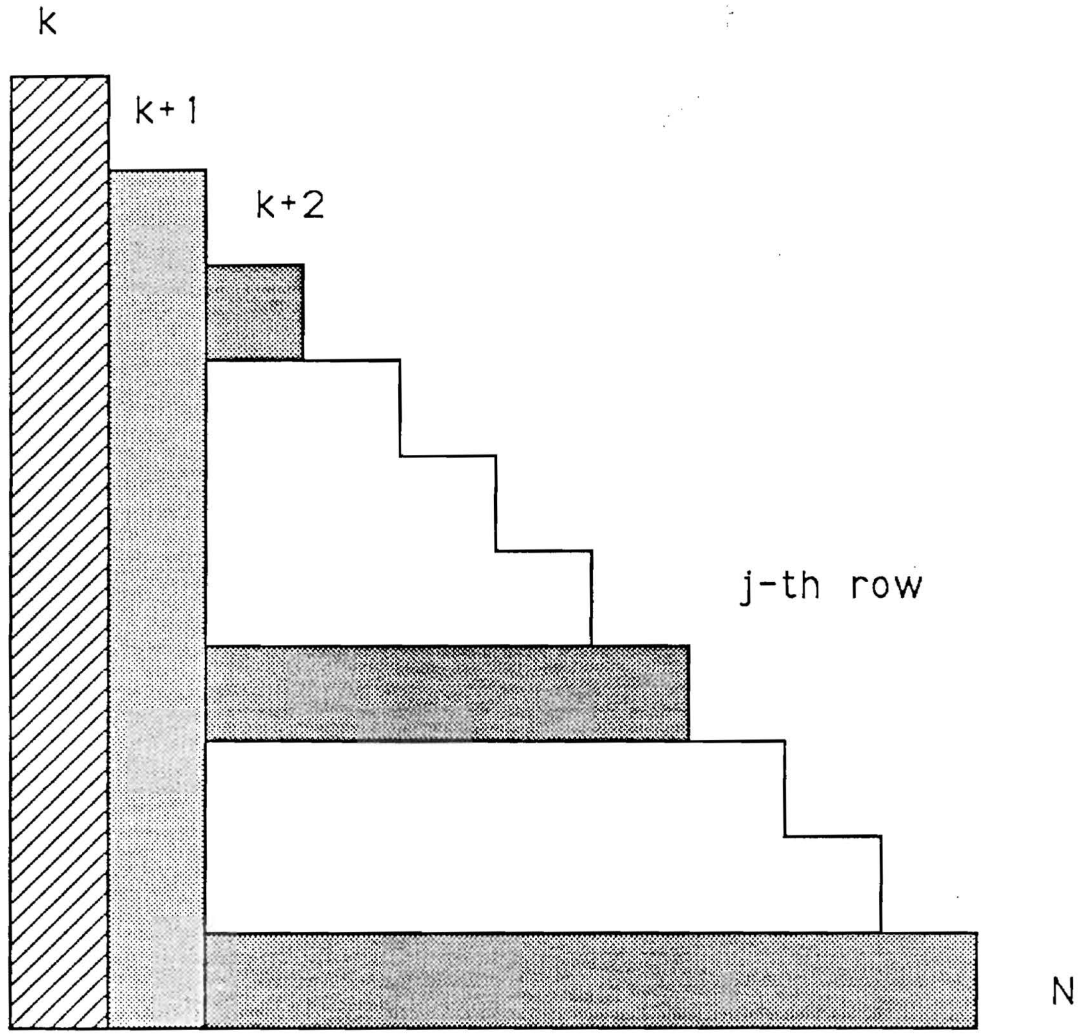
ii) cmod(k)

j = k+2 , N



i) cdiv(k)

$\left.\begin{array}{l}\text{ii) cmod1(k)}\\ \text{iii) cmod2(k, j)}\end{array}\right\}$ cmod(k)



k
k+1
k+2
j-th row
N

Submitted to
NASA Lewis Research Center
by the
Georgia Tech Research Corporation
Georgia Institute of Technology
Atlanta, Georgia 30332

PARALLEL PROCESSING FOR THE ANALYSIS OF TURBO MACHINERY BLADED-DISK ASSEMBLIES

AMOUNT REQUESTED:	$20,024.00
PROPOSAL DURATION:	6 months
REQUESTED STARTING DATE:	June 16, 1989
PRINCIPAL INVESTIGATOR:	Dr. Manohar P. Kamat, Professor
SOCIAL SECURITY NUMBER:	259-84-5901
TELEPHONE:	(404) 894-7439

Manohar P. Kamat
Principal Investigator

R. Dennis Farmer, Contracting Officer
Georgia Tech Research Corporation

Don P. Giddens, Director
School of Aerospace Engineering

William M. Sangster, Dean
College of Engineering

TABLE OF CONTENTS

I. INTRODUCTION

This proposal summarizes progress to date over the past six months and outlines additional tasks that need to be undertaken to eventually provide a finite element analysis capability for Turbo Machinery Bladed Disk Assemblies in a parallel processing environment.

Analysis of aircraft turbo fan engines is computationally intensive. Problems involving aeroelastic stability and response of bladed-disk assemblies in aircraft turbo fan engines are among the most difficult problems encountered. Complications in these studies arise from the small differences between individual blades known as mistuning [1]. Previous researchers have come to believe that the static, flutter and forced response of mistuned turbo machinery blades can be studied by analyzing each blade separately in either a pure bending motion or pure torsional motion. Concurrent (parallel) multiprocessing seems to offer the greatest promise for such an analysis.

The performance limit of modern day computers with a single processing unit has been estimated at 3 billions of floating point operations per second (3 gigaflops). In view of this limit of a sequential unit, performance rates higher than 3 gigaflops can be achieved only through vectorization and/or parallelism as an Alliant Fx-8. Accordingly, the efforts of this critically needed research will be geared towards developing and evaluating parallel finite element methods for static and forced response analysis of multi-degree-of-freedom models of turbo machinery bladed-disk assemblies.

Concurrent processing machines such as the FLEX-32, Alliant FX-80 are multiple instruction, multiple data (MIMD) computers and have the potential for increasing effective calculation speeds by several orders of magnitude. But this potential increase in speed cannot be effectively utilized without the development and implementation of appropriate numerical algorithms which take advantages of the parallel computation features of this new generation of computers. Use of existing algorithms on sequential computers will not realize the full potential of these new MIMD computers and research is needed in the development of parallel structural analysis algorithms for these computers.

Issues involved in implementation of parallel structural algorithms on MIMD supercomputers are much more complex than on current sequential computers and the total hardware/software system must be taken into consideration. The implementation criteria that influence the efficiency of an algorithm include the amount of computation versus the amount of communication in a given problem, the balance of workload among the processors, the communication paths and synchronization delays, and the size of a problem in relation to the number of processors used.

II. CURRENT ACTIVITY AND PROGRESS TO-DATE

Research activity under the grant for the first six months were devoted to familiarizing the two graduate students (Chris Ayers and Chung-Yul Song) with the FLEX 32 multicomputer at the CAE/CAD laboratory of the Georgia Institute of Technology.

The following tasks were completed:

1. Development of a package of linear algebra subroutines similar to LINPACK (but perhaps not as comprehensive) for operation on the FLEX 32;

2. Development and evaluation of a robust QR algorithm suitable for eigenvalue analysis of structural systems with real symmetric matrices;

3. Development and evaluation of an efficient conjugate gradient algorithm for the solution of a system of linear equations of the type commonly encountered in a finite element structural analysis. Such a solution scheme is much superior to the hitherto used $\underline{L}\ \underline{D}\ \underline{L}^T$ factorization scheme in a sequential computational environment;

4. For evaluation of the pay-off from the conjugate gradient algorithm of task 3 above, relative to the best known sequential algorithm, a parellel version of the $\underline{L}\ \underline{D}\ \underline{L}^T$ algorithm was developed. Such an algorithm is also useful for the development of a preconditioned conjugate gradient algorithm (using the element-by-element preconditioned) appropriate for the solution of large scale, ill-conditioned systems of linear and/or nonlinear equations often encountered in large finite element structural systems.

5. Work is continuing on parallelizing an in-house finite element code similar to ADINA [2]. The code currently has an eight-to-twenty one noded isoparametric element suitable for static and transient linear analysis of solids or thick shells and plates.

III. PROPOSED TASKS

1. Future research activity will be focussed primarily on applications of interest to NASA Lewis namely Turbine Blade Dynamics wherein the aerodynamic loads must be treated as nonconservative loads.

2. To better model a typical blade, a new thin shell element with stretching and bending capabilities will be added to the in-house code that is currently being parallelized. This task will involve no new development since the thin shell element from SAP IV [3] will be directly implemented into the code. Some minor modifications may be necessary to accept laminated composite construction.

3. The present static code will be extended to include a capability for linear transient analysis. Again most of this extension will be based on the available capabilities in SAP IV. Future plans will include the implementation of a nonlinear analysis capability with geometric and material nonlinearities.

4. Work will be initiated to extend the present parallel QR algorithm for real matrices to accept complex matrices of the type encountered in aeroelastic analysis of turbine blades.

5. Evaluation of the parallel finite element codes that will be developed will be based on providing speed-ups using the new algorithms on a parallel processing unit such as the Alliant Fx-8 relative to the performance on a single processor of the best known algorithm of the sequential environment. It is anticipated that most of the development of the new algorithms will be made directly on Alliant Fx-80 to minimize the effort required in converting these algorithms from the FLEX-32 to Alliant Fx-80.

IV. REFERENCES

[1]. Kaza, K.R.V. and Kielb, R.E., "Flutter and Response of Mistuned Cascade in Incompressible Flow," AIAA Journal, Vol. 20, No. 2, pp. 1120-1127.

[2]. Bathe, K.J., "ADINA - A Finite Element Program for Automatic Dynamic Incremental Nonlinear Analysis," Report 82448-1, Acoustics and Vibration Laboratory, MIT, Cambridge, MA., 1975.

[3]. Bathe, K.J., Wilson, E.L., and Peterson, F.E., "SAP IV - A Structural Analysis Program for Static and Dynamic Response of Linear Systems," Report EERC 73-11, University of California, Berkeley, April 1974.

PROPOSED BUDGET

June 16, 1989 - December 15, 1989

A.	PERSONAL SERVICES	SPONSOR
	1. Dr. Manohar Kamat, Principal Investigator (Summer)	$ 3,000
	2. 2 Graduate Students	8,750
	TOTAL PERSONAL SERVICES	11,750
B.	FRINGE BENEFITS (25.5% of A.1)	765
E.	INDIRECT COSTS (60% of A+B)	7,509
	TOTAL	$20,024

VI. BIOGRAPHICAL SKETCH

MANOHAR P. KAMAT	Professor of Aerospace Engineering Georgia Institute of Technology
EDUCATION:	**B.S. degree**, Civil Engineering University of Poona (India) 1961 **M.S. degree**, Aerospace Engineering Georgia Institute of Technology **Ph.D. degree**, Engineering Science and Mechanics, Georgia Institute of Technology 1972
PROFESSIONAL AFFILIATIONS:	American Institute of Aeronautics & Astronautics American Society of Civil Engineers American Society of Engineering Education

Dr. Kamat has been active in teaching graduate courses in finite and boundary element methods and is involved in research, which is focused primarily on concurrent multiprocessing for the dynamic analysis of constrained multibody systems, and finite element modeling for linear and nonlinear solid mechanics. Recently, Dr. Kamat completed the direction of a Ph.D. candidate's dissertion on Concurrent Multiprocessors in Computational Mechanics for Constrained Dynamical Systems. Dr. Kamat recently headed a Task Committee of the Aerospace Division of the ASCE, which was funded by AFRPL to prepare a state-of-the- practice report on Methods for Identification of Large Structures in Space. Dr. Kamat's other research interests include application and coupling of finite and boundary element techniques for stress analysis and optimization of conventional and laminated composite structures.

Before joining the Georgia Institute of Technology in September 1985, Dr. Kamat was Professor of Engineering Science and Mechanics of the Virginia Polytechnic Institute for the past twelve years, where he taught undergraduate and graduate courses in strength of materials, dynamics, numerical methods and optimization and concentrated his research efforts in two major areas: (1) development of algorithms for nonlinear transient analysis of structures and (2) crashworthiness studies of aircraft and automobiles, with a minor emphasis on stability and failure analysis of laminated composite plates and shells.

Dr. Kamat has published over forty archival journal articles and a textbook on structural optimization. In the past he has served as the Associate Editor of the AIAA Journal and is currently on the Editorial Board and the Publication Committee of the ASCE's new Journal of Aerospace Engineering.

VII. STAFFING/FACILITIES/RESOURCES

The proposed work will be led by Dr. Manohar P. Kamat, Professor of Aerospace Engineering, and by his graduate students. Dr. Kamat has been actively involved in the application of the optimization techniques for the solution of problems in crash dynamics, nonlinear structural analysis, and design and control, and lately in concurrent multiprocessing of constrained multibody systems. Student participation would involve one or two graduate research assistants at the masters and/or doctoral levels. Clerical support will be provided part-time through the existing Georgia Tech staff.

The research will utilize FLEX/32 multicomputer with twenty processors located at the CAE/CAD Laboratory of Georgia Tech as well as Alliant FX-80 located at the NASA Lewis Research Center. These computers provide representative capabilities to evaluate candidate algorithms on appropriate test problems. In addition, an Electrical Engineering group at Georgia Tech is currently building a parallel computer targeted for thousands of processors to be used for currently building a parallel computer targeted for thousands of processors to be used for dynamics/control applications. An initial 64 processor system is now operational and is soon expected to be available for application studies. Furthermore, a Convex Sequent machine is also available at Tech with plans for the purchase of a BBN Buterfly in the not too distant future.

The principal investigator has currently no other support from other agencies in the area of parallel processing.

(Grants/Cooperative Agreements)

A. The grantee certifies that it will provide a drug-free workplace by:

(a) Publishing a statement notifying employees that the unlawful manufacture, distribution, dispensing, possession or use of a controlled substance is prohibited in the grantee's workplace and specifying the actions that will be taken against employees for violation of such prohibition;

(b) Establishing a drug-free awareness program to inform employees about --

(1) The dangers of drug abuse in the workplace;

(2) The grantee's policy of maintaining a drug-free workplace;

(3) Any available drug counseling, rehabilitation, and employee assistance programs; and

(4) The penalties that may be imposed upon employees for drug abuse violations occurring in the workplace;

(c) Making it a requirement that each employee to be engaged in the performance of the grant be given a copy of the statement required by paragraph (a);

(d) Notifying the employee in the statement required by paragraph (a) that, as a condition of employment under the grant, the employee will --

(1) Abide by the terms of the statement, and

(2) Notify the employer of any criminal drug statute conviction for a violation occurring in the workplace no later than five days after such conviction;

(e) Notifying the agency within ten days after receiving notice under subparagraph (d) (2) from an employee or otherwise receiving actual notice of such conviction;

(f) Taking one of the following actions, within 30 days of receiving notice under subparagraph (d) (2), with respect to any employee who is so convicted --

(1) Taking appropriate personnel action against such an employee, up to and including termination; or

(2) Requiring such employee to participate satisfactorily in a drug abuse assistance or rehabilitation program approved for such purposes by a Federal, State, or local health, law enforcement, or other appropriate agency;

(g) Making a good faith effort to continue to maintain a drug-free workplace through implementation of paragraphs (a), (b), (c), (d), (e) and (f).

B. The grantee shall insert in the space provided below the site(s) for the performance of work done in connection with the specific grant:

Place of Performance (Street address, city, county, state, zip code)

225 North Avenue

Atlanta, Fulton County, GA. 30332

Signature of Responsible University Official and Date:

15 March 1989

Typed Name and Title: R. Dennis Farmer, Contracting Officer

Title/Identification of Applicable Research Proposal:

Parallel Processing for the Analysis of Turbo Machinery Bladed-Disk Assemblies